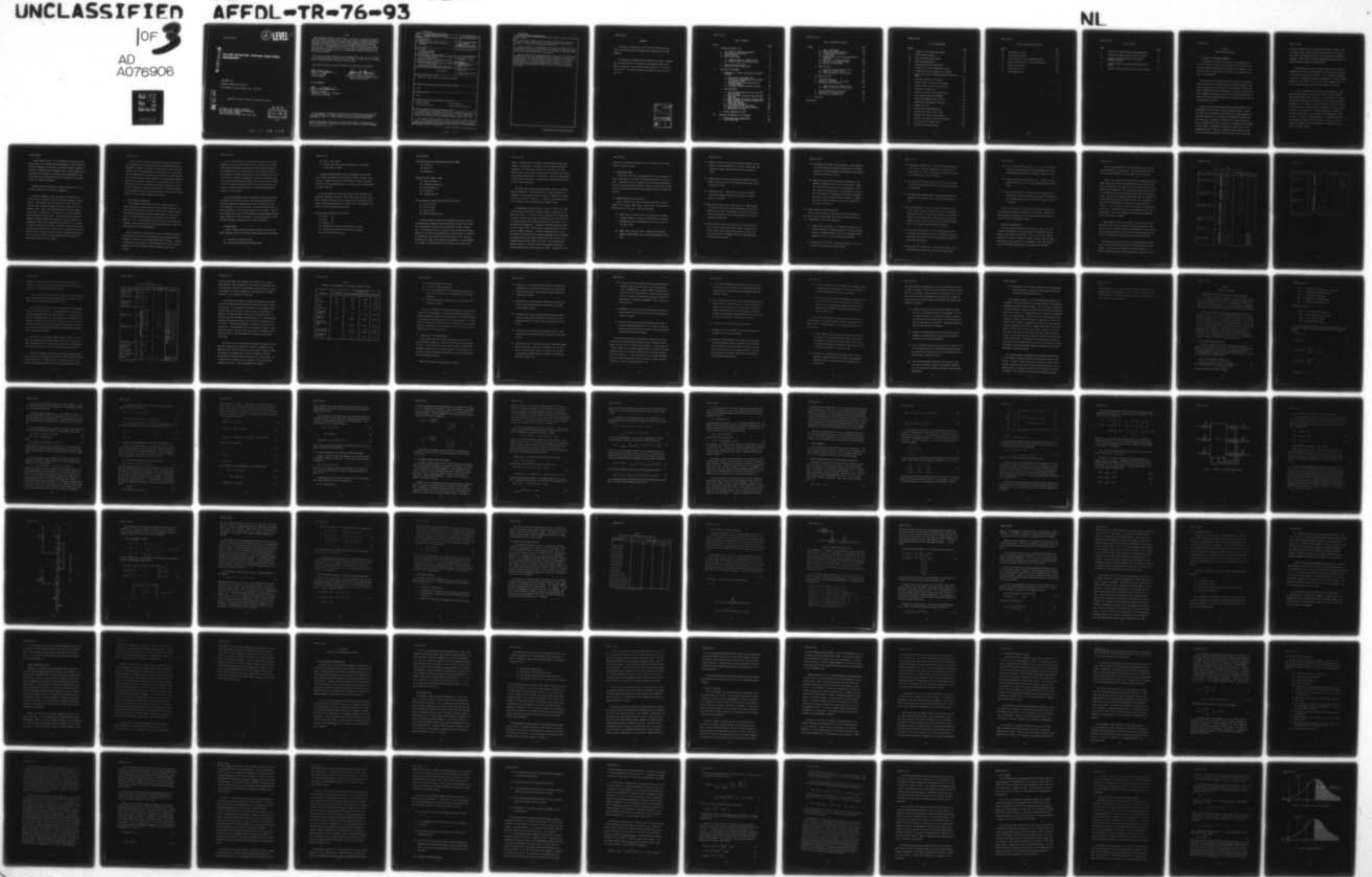
AD-A076 906   AIR FORCE FLIGHT DYNAMICS LAB WRIGHT-PATTERSON AFB OH   F/G 1/3
FAILURE DETECTION THROUGH FUNCTIONAL REDUNDANCY.(U)
SEP 76 P S MAYBECK
UNCLASSIFIED   AFFDL-TR-76-93   NL

1 OF 3

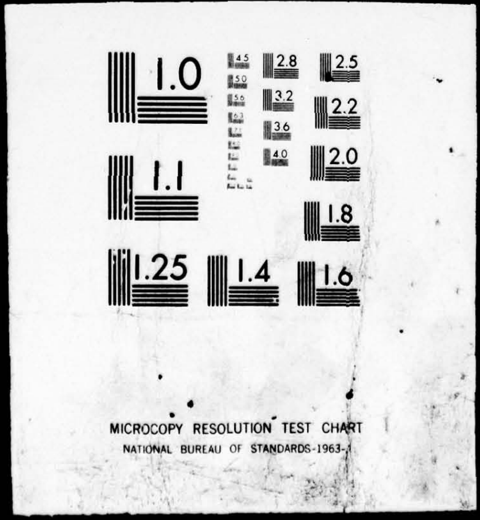
AD
A076906



SIFIED

1 OF 3

AD
A076906



MICROCOPY RESOLUTION TEST CHART
NATIONAL BUREAU OF STANDARDS-1963-A

AFFDL-TR-76-93

FAILURE DETECTION THROUGH FUNCTIONAL REDUNDANCY

SEPTEMBER 1976

TECHNICAL REPORT AFFDL-TR-76-93

Final Report for Period January 1972 - July 1973

Approved for public release; distribution unlimited.

AIR FORCE FLIGHT DYNAMICS LABORATORY
AIR FORCE WRIGHT AERONAUTICAL LABORATORIES
AIR FORCE SYSTEMS COMMAND
WRIGHT-PATTERSON AIR FORCE BASE, OHIO 45433

NOTICE

When Government drawings, specifications, or other data are used for any purpose other than in connection with a definitely related Government procurement operation, the United States Government thereby incurs no responsibility nor any obligation whatsoever; and the fact that the government may have formulated, furnished, or in any way supplied the said drawings, specifications, or other data, is not to be regarded by implication or otherwise as in any manner licensing the holder or any other person or corporation, or conveying any rights or permission to manufacture, use, or sell any patented invention that may in any way be related thereto.

This report has been reviewed by the Information Office (OI) and is releasable to the National Technical Information Service (NTIS). At NTIS, it will be available to the general public, including foreign nations.

This technical report has been reviewed and is approved for publication.

Peter S. Maybeck
PETER S. MAYBECK
Project Engineer

James W. Morris
JAMES W. MORRIS, Acting Chief
Control Systems Development Branch
Flight Control Division

FOR THE COMMANDER

Morris A. Ostgaard
MORRIS A. OSTGAARD
Assistant for Research & Technology
Flight Control Division

"If your address has changed, if you wish to be removed from our mailing list, or if the addressee is no longer employed by your organization please notify AFFDL/FGL ,W-PAFB, OH 45433 to help us maintain a current mailing list".

Copies of this report should not be returned unless return is required by security considerations, contractual obligations, or notice on a specific document.

AIR FORCE/56780/24 October 1979 — 300

UNCLASSIFIED

SECURITY CLASSIFICATION OF THIS PAGE (When Data Entered)

14 **REPORT DOCUMENTATION PAGE**	READ INSTRUCTIONS BEFORE COMPLETING FORM
1. REPORT NUMBER AFFDL-TR-76-93 ✓ 2. GOVT ACCESSION NO.	3. RECIPIENT'S CATALOG NUMBER 9
4. TITLE (and Subtitle) 6 FAILURE DETECTION THROUGH FUNCTIONAL REDUNDANCY,	5. TYPE OF REPORT & PERIOD COVERED FINAL REPORT, JANU~~ARY~~ 1972— JUL~~Y~~ 1973
	6. PERFORMING ORG. REPORT NUMBER
7. AUTHOR(s) 10 DR. PETER S. MAYBECK ~~1st Lieutenant, USAF~~	8. CONTRACT OR GRANT NUMBER(s)
9. PERFORMING ORGANIZATION NAME AND ADDRESS Air Force Flight Dynamics Laboratory (AFFDL/FGL) Air Force Systems Command Wright-Patterson Air Force Base, Ohio 45433	10. PROGRAM ELEMENT, PROJECT, TASK AREA & WORK UNIT NUMBERS 16 19870250 17 02 62201F
11. CONTROLLING OFFICE NAME AND ADDRESS Air Force Flight Dynamics Laboratory (AFFDL/FGL) Air Force Systems Command Wright-Patterson Air Force Base, Ohio 45433	11 12. REPORT DATE Sep~~tember~~ 1976 ✓
	13. NUMBER OF PAGES 192
14. MONITORING AGENCY NAME & ADDRESS(if different from Controlling Office) 12 202	15. SECURITY CLASS. (of this report) UNCLASSIFIED
	15a. DECLASSIFICATION/DOWNGRADING SCHEDULE NA

16. DISTRIBUTION STATEMENT (of this Report)

Approved for public release; distribution unlimited

17. DISTRIBUTION STATEMENT (of the abstract entered in Block 20, if different from Report)

Same

18. SUPPLEMENTARY NOTES

None

19. KEY WORDS (Continue on reverse side if necessary and identify by block number)

Redundant Systems	System Reliability
Digital Flight Control	Failure Detection Algorithms

20. ABSTRACT (Continue on reverse side if necessary and identify by block number)

A high degree of reliability in control data instrumentation is vital to the Air Force's the mission of the Air Force, and this objective is typically accomplished by comparing the outputs of redundant system components. However, the weight, volume, and cost penalties of such equipment redundancy can be substantial.

Data systems already onboard the aircraft--the air data computer, inertial system, attitude and heading reference system, and the rate gyros and accelerometers of the automatic flight control system--provide functionally ~~realted~~ related (Over)K

DD FORM 1 JAN 73 1473 EDITION OF 1 NOV 65 IS OBSOLETE

SECURITY CLASSIFICATION OF THIS PAGE (When Data Entered)

012 070   LB

BLOCK 20 CONT.

data. This form of inherent functional redundancy among sensor signals can be exploited rather than resorting exclusively to hardware duplication to achieve the desired level of data system reliability.

A major objective of this study was ~~effort has been~~ to demonstrate both ~~not only~~ concept feasibility and ~~, but~~ the extent of performance capabilities of a functional redundancy failure detection algorithm. Thus, in a performance improvement program, both missed alarms and false alarms were minimized.

The other main goal was ~~of the work has been~~ to develop the failure detection algorithm and an associated performance analysis program into a systematic and flexible design tool, which ~~This tool~~ would then be used to generate an eventual implementation of a tuned and optimized software package for a particular on-board application of functional redundancy for failure detection. Such an ~~useful~~ engineering tool was ~~has been~~ developed, and it promises to assist substantially in converting the performance potential of functional redundancy into reality.

FOREWORD

This report was prepared by the Air Force Flight Dynamics Laboratory, Wright-Patterson Air Force Base, Ohio, under work unit number 19870250.

The investigator and author of this report was Dr. Peter S. Maybeck, who, during the period of investigation, was employed at the Flight Dynamics Laboratory, in the Control Elements Branch of its Flight Control Division. The report covers work performed from January 1972 to July 1973.

TABLE OF CONTENTS

SECTION PAGE

I BACKGROUND AND OBJECTIVES 1

1. THE CONCEPT OF FUNCTIONAL REDUNDANCY 1
2. BASIC ALGORITHM FORMULATION 4
3. SYSTEMS EMPLOYED 5
4. MAJOR FAILURE MODES 9

4.1 SUDDEN FAILURES WITH SUDDEN EFFECTS 9
4.2 SUDDEN FAILURES WITH DRIFTING EFFECTS 11

5. RESULTS OF PREVIOUS STUDY 13
6. OBJECTIVES OF THIS INVESTIGATION 21

II THEORETICAL DEVELOPMENT 29

1. FUNDAMENTALS OF KALMAN FILTERING AND LIKELIHOOD FUNCTIONS 29

1.1 THE DISCRETE-TIME KALMAN FILTER 29
1.2 DISCRETE-TIME REPRESENTATION OF CONTINUOUS-TIME DYNAMIC SYSTEMS 32
1.3 EXTENDED KALMAN FILTER FOR SYSTEM WITH NONLINEAR DYNAMICS 34
1.4 LIKELIHOOD FUNCTION STATISTICAL TESTING 35

2. MODEL REFERENCES 39

2.1 MODEL REFERENCE RELATING INS ATTITUDES AND AFCS BODY RATES 39
2.2 MODEL REFERENCE FOR AHRS ATTITUDES AND AFCS BODY RATES 44
2.3 MODEL REFERENCE FOR AIRCRAFT VERTICAL MOTION 44
2.4 MODEL REFERENCE FOR INDICATED AIRSPEED AND NORMAL ACCELERATION 48
2.5 OTHER MODEL REFERENCES 49
2.6 FAILURE DETECTION AND ISOLATION 50
2.7 MODIFICATIONS TO THE MODEL REFERENCES 52
2.8 ADAPTATION TO FAILURES 57

3. OPTIMAL COMBINATION OF DATA 59

III PRACTICAL APPLICATION OF THE TECHNIQUE 62

1. SYSTEM DESIGN AND IMPLEMENTATION 62
2. USE OF DESIGN TOOL 63

TABLE OF CONTENTS (Concluded)

SECTION		PAGE
	2.1 BASIS OF COMPARISON	66
	2.2 ESTABLISHING SENSOR STATISTICS	69
	2.3 SOFTWARE INPUTS AND OUTPUTS	72
	2.4 FILTER TUNING	83
	2.5 THRESHOLDS AND TIME-TO-FAILURE-DECLARATION PARAMETERS	84
	2.6 SENSITIVITY TO AIRCRAFT MANEUVERS	92
	2.7 SENSITIVITY TO INSTRUMENT BIASES	96
	2.8 PREFLIGHT INITIALIZATION	97
	3. MODES OF USAGE	98
	3.1 INTEGRATED FAILURE DETECTION SYSTEM	99
	3.2 MEANS OF DECLARING FAILURES	100
	3.3 LOGIC ADAPTATION TO FAILED SENSORS	101
IV	EXPERIMENTAL RESULTS	104
	1. EXPERIMENTS CONDUCTED	104
	2. BASELINE OF PERFORMANCE - NO FAILURES	106
	3. FAILURE DETECTION CAPABILITY	124
	3.1 SUDDEN FAILURES WITH SUDDEN EFFECTS	124
	3.2 SUDDEN FAILURES WITH DRIFTING EFFECTS	141
	4. LESSENING SENSITIVITY TO ROLL RATE	172
	5. SENSITIVITY TO INSTRUMENT BIASES	178
	6. VERIFICATION OF SIMULATED DATA	188
V	CONCLUSION	191
BIBLIOGRAPHY		192

LIST OF ILLUSTRATIONS

FIGURE		PAGE
1	Schematic of Attitude Dynamics Model	43
2	Vertical Motion Dynamics	45
3a	White Noise Corruption; No Bias	52
3b	Model Incorporating Bias	53
4	Probability of Failure Declaration	86
5	Probability of Failure Declaration vs b/σ_t	88
6	Probability of Failure Detection vs $b/(TAe)^{V2}$	89
7	Sensitivity of Roll Likelihood Function to Rapid Roll Rates	93
8	Likelihood Functions - No Failures	107
9	Excessive Noise in Static Pressure Output	127
10	Excessive Noise in Pitot Pressure Output	130
11	Bent Angle of Attack Vane	131
12	Recovery of Vertical Likelihood Functions	135
13	Noisy Angle-of-Attack Potentiometer	137
14	Normal Accelerometer Pickoff Failure	140
15	INS Vertical Accelerometer Float Leak	142
16	Clogged Pitot Line	146
17	INS Vertical Gyro Torquer Failure	148
18	INS Gyro Float Leak in Level Flight	151
19	INS Gyro Float Leak in Turn-Attitudes	155
20	INS Gyro Float Leak in Turn-Vertical Channel	158
21	Loss of Cutoff for Vertical Gyro	162
22	Directional Gyro Servo Failure	165

LIST OF ILLUSTRATIONS (Concluded)

FIGURE		PAGE
23	Yaw Rate Gyro Failure	166
24	Roll Rate Gyro Failure	173
25	Rate Gyro Failure; Q_{11} = Twice Normal Value	176
26	Rate Gyro Failure; Q_{11} = Ten Times Normal Value	179
27	Static Pressure Bias	182
28	Pitot Pressure Bias	185
29	AHRS Compass Bias	189

LIST OF TABLES

TABLE		PAGE
I	Detection of Sudden Failures with Sudden Effects	15
II	Detection of Sudden Failures with Drift Effects	18
III	Computer Time Requirements of the Internal Redundancy Method	20
IV	Sensor Failures and Corresponding Abnormal Error Signals	51
V	Largest Likelihood Function Values with No Failures	123

SECTION I

BACKGROUND AND OBJECTIVES

1. THE CONCEPT OF FUNCTIONAL REDUNDANCY

A high degree of reliability in control data instrumentation is vital to the mission of USAF, and this objective is typically accomplished by comparing the outputs of redundant system components. However, the weight, volume, and cost penalties of such equipment redundancy can be substantial.

Data systems already onboard the aircraft - the air data computer, inertial system (free or aided), attitude and heading reference system, and the rate gyros and accelerometers of the automatic flight control system - provide functionally related data. This form of inherent functional redundancy among sensor signals can be exploited rather than resorting exclusively to hardware duplication to achieve the desired level of data system reliability.

To date, such functional redundancy has not been employed in the development of fault tolerant or high reliability systems. Instead, the reliability of individual system components has been improved, and then these components are incorporated redundantly with some form of comparison logic to generate a reliable signal. Such comparisons require a minimum of two signals to indicate a discrepancy, and a minimum of three signals to determine the appropriate signal level if a discrepancy does exist. Thus, for a system to operate normally in the face of a single sensor failure, that sensor must exist in triplicate so that the comparison logic can isolate (and remove) the faulty signal. It is desirable for the more critical systems to be able to tolerate two failures, necessitating another level of redundancy, or quadruplet components. Such a level of hardware redundancy becomes prohibitive as systems become more complex and sophisticated.

A viable alternative would be to replace some levels of hardware redundancy with the functional redundancy that exists among the outputs of the different sensors in the aircraft. Thus, functional redundancy is viewed as a complement to equipment redundancy for an overall system, providing the same level of reliability with fewer components than required if subsystem outputs were not correlated with one another.

It is also a complement to, rather than a replacement for, other means of fault detection. Some types of failures are more appropriately handled by these other techniques. As envisioned in this report, the functional redundancy algorithm might have an iteration rate of approximately 5 Hz. A number of iterations might be required to declare a failure and isolate the failed signal, resulting in a time period on the order of one second before a failed signal might be removed from the overall data system. Such a response time would probably not be sufficient for safety of flight parameters. The iteration rate would be increased; but since most signals would not require a faster response time, the benefits would become marginal compared to the increase in computer loading.

Certain types of errors, such as deterministic biases and scale factor errors, can be readily evaluated by means of filtering and compensation techniques on individual or redundant identical components. Functional redundancy might be employed in part to detect which signal is biased out of tolerance, but estimating and compensating for the actual bias value is achieved more easily by comparison of the signal to that of an identical component.

Similarly, many hard failures are readily detected by built-in-test (BIT) capabilities of individual components.

Functional redundancy is most applicable to the detection of failures that are currently isolated by comparing signals of identical sensors. These might be hard (or catastrophic) signal failures or "soft" failures in which the signal slowly drifts away from the true parameter value. Rather than comparing duplicate signals, though, a sensor signal is compared to an estimate of its value generated from other functionally related signals. These functional relationships encompass kinematic differential equations, as well as geometric and aerodynamic relations that characterize aircraft motion. For example, an inertial system indicates angular orientation of the vehicle, while rate gyros associated with the flight control system measure the vehicle angular rates. A kinematic relationship exists between these quantities, and such a functional relationship allows the correlation of data taken from the two sensor systems.

Actually, the pilot and crew have been required to perform such cross correlation in verifying the validity of independent subsystem measurements. However, as aircraft incorporate more sophisticated and extensive avionics, multiple mode or mission capabilities, and higher speed and meaneuverability, especially in the case of single-seat vehicles, the time the crew is able to devote to such performance monitoring becomes very restricted. Therefore, it is essential that as much information as possible be automatically digested, interpreted, and presented to the crew in a usable and concise form. Functional redundancy can be incorporated into the data system design to provide such reliable capability with a minimum of equipment duplication.

2. BASIC ALGORITHM FORMULATION

The functional relationships provide the system equations of a model reference for the failure detection and isolation technique. The appropriate sensor signals are used as "inputs" to this model reference, the functional relationships thereby generating model reference "outputs." By comparing these outputs to the measured values of these quantities, i.e., signals generated by other sensors, error signals are produced. These are then fed back through appropriate gains so that the model reference tracks the measurements.

When the functional relationships are linear differential equations and the statistics of noises and uncertainties are adequately modelled as Gaussian, the Kalman filter provides such a model reference. Moreover, if the dynamics are nonlinear, an extended Kalman filter that linearizes about the most recent estimate of nominal parameter values can be utilized.

Sensor failure detection is achieved by monitoring the individual components of the sequence of residuals, the individual error signals generated by differencing a measured output and the model reference's best estimate of what its value should be. With no sensor failures, this sequence of residuals should possess certain characteristics, such as being white, zero-mean, and Gaussian. A consistent departure from such a characterization would indicate a fault, and the specific manner in which this departure manifests itself in the residual sequence can be used to isolate the particular fault involved (at least partially isolate it, if not totally).

A logical and effective means of discerning such departures would be through the use of the statistical detection theory method of observing the magnitude of appropriately defined likelihood functions. If the magnitude of a certain residual is consistently higher than normal, the magnitude of the likelihood function also increases. When its value surpasses some preselected threshold for acceptable behavior under normal conditions, a fault is declared. By noting the pattern of such threshold passings, the exact cause can (often) be deduced.

3. SYSTEMS EMPLOYED

In order to demonstrate the performance capabilities of the functional redundancy concept, the following sensor systems were considered:

(1) the inertial navigation system

(2) the attitude and heading reference system

(3) the air data system

(4) the rate gyros and normal accelerometer of the automatic flight control system.

The concept can be applied to other measurement systems as well, such as those associated with the propulsion system or external navigation aids. However, the scope of this work was confined to the above systems to yield a concerted effort in an area partially investigated by a previous study [0].

There are eighteen (18) individual signals to be utilized in the functional redundancy algorithm. These signals would be sent to a computer interface which would provide sampling and A/D conversion of the signals, yielding algorithm inputs in usable form. The individual signals from the four measurement systems are:

From the Inertial Navigation System (INS):

(1) Pitch (θ)

(2) Roll (ϕ)

(3) Heading (ψ)

(4) Acceleration along local horizontal axis x_h (a_{xh})

(5) Acceleration along local horizontal axis y_h (a_{yh})

(6) Vertical acceleration (a_{zh})

From the Attitude and Heading Reference System (AHRS):

(1) Pitch (θ')

(2) Roll (ϕ')

(3) Heading (ψ')

From the Air Data Computer (ADC):

(1) Angle of attack (α)

(2) Indicated airspeed (v_i)

(3) True airspeed (v_a)

(4) Altitude (h)

(5) Altitude rate ($\dot{h}$)

From the Automatic Flight Control System (AFCS) sensors:

(1) Pitch rate (w_y)

(2) Roll rate (w_x)

(3) Yaw rate (w_z)

(4) Normal acceleration (a_z)

The three angles from the INS are available from gimbal resolvers or from the gyros themselves; the accelerations are taken from the platform accelerometers, and are thus coordinatized in local horizontal axes. (The exact definition of x_h and y_h in the horizontal plane would depend on the inertial system mechanization, and can actually be defined for convenience since any choice would be related to what is actually available by a simple, known rotation transformation.) The displacement gyro assembly of the AHRS provides its indication of the three Euler angles. To generate its five signals, the ADC processes inputs from various air data sensors: the angle of attack probe and associated transmitter, the temperature probe and its transmitter, the static pressure source, and the pitot pressure source (the latter two typically being coupled pneumatically to the computer). Finally, the AFCS sensors provide signals proportional to the body rates and normal acceleration in vehicle body coordinates.

The signals described above are the nominal inputs to the failure detection algorithm. It may be beneficial to input an indication of commanded or actual control surface positions for adaptability purposes, as will be discussed in Section III.7. However, the basic description of the system will first consider only the nominal inputs.

The measurement systems employed are found in virtually all modern aircraft, and thus the failure detection concept is applicable to any particular vehicle. To assure a realistic evaluation of the technique, a particular aircraft was chosen to represent typical applications; the F-4 chosen because of the relative availability of data about its performance and instrumentation. A previously developed simulation model of the F-4 vehicle and its various sensor subsystems [0] was utilized in the first phases of analysis. Since any simulation model is a simplification of the real world environment, subsequent analysis replaced the simulation with actual data recordings from test aircraft, with simulated sensor failures added to real data. This second phase of analysis provided as realistic a means of performance evaluation as possible without actual sensor failures in flight. Essentially, it was conducted to corroborate the conclusions of the analysis based on the aircraft simulation program.

4. MAJOR FAILURE MODES

A substantial effort was conducted previously [ ] to delineate the major failure modes of the sensor instrumentation onboard a typical aircraft. These modes were then included in the simulation program used to evaluate failure detection performance (a portion of this effort included the revision and modification of the program to provide a better simulation).

4.1 SUDDEN FAILURES WITH SUDDEN EFFECTS

Certain failures affect sensor measurements directly, so that a sudden failure causes a sudden effect. Many failures involving the air data system are of this type. These would include:

(1) Sudden leak in the static line: this would cause a sudden erroneous measure of altitude, altitude rate, and indicated airspeed, the error being detectable during any portion of the flight regime.

(2) Sudden leak in the pitot line: indicated airspeed would undergo a sudden change, this error being detectable at any time.

(3) Sudden increase in noise of static pressure transducer output: would cause noisy measurements of altitude, altitude rate, and indicated airspeed, being detectable during any portion of flight.

(4) Sudden increase in noise of pitot pressure transducer output: would similarly cause a noisy indicated airspeed signal, being evident at any time.

(5) Tachometer failure: would result in the loss of altitude rate, being detectable only when the aircraft is either ascending or descending.

(6) Bent angle-of-attack vane: would result in a sudden increase in the bias of the angle-of-attack measurement, and would be detectable during any flight regime; this would also adversely affect the computed indicated airspeed generated in the failure detection algorithm.

(7) Sudden increase in noise of angle-of-attack output potentiometer: angle-of-attack signal would undergo a sudden increase in noise level, occurring during any portion of flight; this, too, would corrupt the computed indicated airspeed developed in the detection algorithm.

(8) AFCS normal accelerometer pickoff failure: would immediately affect the computed indicated airspeed created by the model reference of the failure detection algorithm, and this effect should be noticeable at any time.

(9) Sudden float leak of an INS vertical accelerometer: would generate a sudden error in the vertical acceleration signal, and a slow drifting effect on INS altitude outputs as well. Only the vertical accelerometer is used directly in the present failure detection algorithm--had the INS accelerometers been used to check the AFCS normal accelerometer, a sudden failure of an INS accelerometer would yield a sudden error in this signal correlation.

4.2 SUDDEN FAILURES WITH DRIFTING EFFECTS

Certain types of failures do not directly affect measurements, so that their results are not sudden, but drifting, erroneous signal levels. Failure modes of this form encompassed:

(1) Clogged static line to ADC: the altitude, altitude rate, and indicated airspeed will become erroneous if the vehicle changes altitude or airspeed (neither of which can be changed instantaneously, so this is in fact a drifting type effect).

(2) Clogged pitot line to ADC: indicated airspeed will drift off true value if the vehicle changes airspeed.

(3) Jammed angle-of-attack vane: the ADC angle-of-attack signal will exhibit a "drifting" type error as the true angle of attack varies in flight. (The simulation program was incapable of producing this type of failure.)

(4) INS vertical gyro torquer failure: drifting of pitch and/or roll attitude indications would result, eventually corrupting all INS outputs.

(5) INS heading gyro torquer failure: heading measurement would undergo a drift, and other INS outputs would be affected in time.

(6) INS gyro float leak: this, or any other failure that would cause a center of gravity shift, will result in acceleration-induced gyro drifts; in level flight the gravity induced drift affects the INS heading output, while during a turn both the heading and pitch indications of the INS are affected; eventually, all INS outputs would be adversely affected.

(7) Vertical gyro servo failure: loss of slaving causes a drift in the AHRS roll indication, so this is detectable only during relatively level flight, since the slaving loop is turned off during high rate maneuvers.

(8) Directional gyro servofailure: loss of slaving yields a drift in the AHRS heading signal during relatively level flight.

(9) Failure of cutoff for vertical gyro: if the servo cutoff system fails to operate, the AHRS bank measurement drifts when a nongravitational acceleration is present, as during a turn.

(10) Failure of cutoff for directional gyro: similarly the AHRS heading indication will drift under nongravitational accelerations.

(11) AFCS rate gyro failure: during a period of changing vehicle orientation, a rate gyro failure (as, a pickoff failure yielding no output from the gyro) will cause the rate indication to be erroneous.

There are also drifting failures that cause drifting effects, but the simulation program does not account for these modes. Nevertheless, the preceding two categories of failure modes should indicate the viability of this failure detection concept.

5. RESULTS OF PREVIOUS STUDY

A nominal approach trajectory involving level flight, final turn, pitchover, and descent was utilized as a means of evaluating the performance of this detection technique. First, a set of runs were conducted with no failures simulated, in order to specify bounds on likelihood function values under normal circumstances. These then would be used as the thresholds beyond which a failure would be declared.

Once this was accomplished, the same trajectory was flown with failures simulated at various points. The time to detection of the failure was then recorded, as was the time to any false alarm.

Table I, taken from the previous study [0], summarizes the experimental results for detection of sudden failures with sudden effects. The first column lists the type of failure simulated along with the expected failure indications. The second column denotes the portion of the trajectory (level, turn, or descent) during which the failure occurred. The third column specifies the actual means of simulating the failures and the magnitudes of these failures. For those cases in which different magnitudes are involved, the notation used is b = a bias, σ = the 1σ value to specify the strength of a Gaussian noise source, and ε = coefficients of gyro or accelerometer errors linear in acceleration. The last two columns indicate the time to detection and/or false alarms in seconds.

Sudden changes in bias (due to leaks) or noise level are readily detected for static and pitot pressure sources, the detection being more sensitive to altitude rate than altitude effects of such a failure (static pressure failures did not affect indicated airspeed due to an error in the simulation program, which has since been rectified).

Both bias and noise type failures on the angle-of-attack measurement signal were detected for sufficiently large magnitude failures. However, a false alarm on airspeed was consistently obtained. The angle-of-attack value would influence the vertical Kalman filter and

TABLE I

DETECTION OF SUDDEN FAILURES WITH SUDDEN EFFECTS

Failure: Erroneous Output	Location*	Simulation	Time Before Detection (seconds)	False Alarms
Massive leak in static line: Altitude; Altitude rate	Turn	$b_p = b_p + 50$	∞ , ∞	--
		$b_p = b_p + 100$	∞ , 1.4	--
		$b_p = b_p + 200$	∞ , 0.4	--
		$b_p = b_p + 400$	0.8, 0.2	--
		$b_p = b_p + 800$	0.2, 0.2	--
Massive leak in pitot line: Indicated airspeed	Turn	$b_p^* = b_p^* + 75$	8.6	--
		$b_p^* = b_p^* + 150$	1.0	--
		$b_p^* = b_p^* + 300$	0.4	--
		$b_p^* = b_p^* + 600$	0.2	--
		$b_p^* = b_p^* + 1200$	0.2	--
Excessive noise in static pressure output altitude: Altitude rate	Turn	$\sigma_p = 50$	∞ , 1.4	--
		$\sigma_p = 100$	∞ , 0.4	--
		$\sigma_p = 200$	1.3, 0.4	--
		$\sigma_p = 400$	0.6, 0.4	--
		$\sigma_p = 800$	0.4, 0.4	--
Excessive noise in pitot pressure output: Indicated airspeed	Turn	$\sigma_p^* = 75$	∞	--
		$\sigma_p^* = 150$	∞	--
		$\sigma_p^* = 300$	1.2	--
		$\sigma_p^* = 600$	0.4	--
		$\sigma_p^* = 1200$	0.2	--
Tachometer failure: Altitude rate	Descent	$\sigma_h = 100$	1.2	--
Bent angle-of-attack vane: Angle of attack	Turn	$b_\alpha = b_\alpha + 0.0075$	∞	Airspeed (8.6 s)
		$b_\alpha = b_\alpha + 0.0150$	∞	Airspeed (1.2 s)
		$b_\alpha = b_\alpha + 0.030$	∞	Airspeed (0.6 s)
		$b_\alpha = b_\alpha + 0.060$	1.4	Airspeed (0.4 s)
		$b_\alpha = b_\alpha + 0.12$	0.4	Airspeed (0.2 s)

TABLE I (Concluded)

Failure: Erroneous Output	Location*	Simulation	Time Before Detection (seconds)	False Alarms
Noisy potentiometer: Angle of attack	Turn	$\sigma_\alpha = 0.0075$	∞	--
		$\sigma_\alpha = 0.0150$	∞	--
		$\sigma_\alpha = 0.0300$	4.0	Airspeed (7.2 s)
		$\sigma_\alpha = 0.060$	2.4	Airspeed (0.8 s)
		$\sigma_\alpha = 0.120$	0.8	Airspeed (0.4 s)
Normal accelerometer pickoff failure: Airspeed	Turn	$\xi^{na} = -1$	0.2	--
		$\sigma_{na} = 0$		
INS accelerometer float leak: Vertical acceleration	Turn	$\bar{\xi}^a_{33} = \bar{\xi}^a_{31} = 0.0025$	∞	--
		$\bar{\xi}^a_{33} = \bar{\xi}^a_{31} = 0.005$	∞	--
		$\bar{\xi}^a_{33} = \bar{\xi}^a_{31} = 0.01$	∞	--
		$\bar{\xi}^a_{33} = \bar{\xi}^a_{31} = 0.02$	∞	--
		$\bar{\xi}^a_{33} = \bar{\xi}^a_{31} = 0.04$	∞	--

* Level, turn, and descent locations in the landing approach correspond to Points 1, 2, 3, respectively, in Figure 13.

airspeed check portions of the model reference (see paragraph 2 of Section II), and so such an alarm could be expected, but after, rather than before, the appropriate failure detection.

It was surmised in the report that the vertical acceleration errors were not detected because the thresholds were set at too high a magnitude, though not substantiated by further analysis.

Table II portrays the results of the experiments involving sudden failures with drifting effects. The clogged static line was detected, but the corresponding pitot line failure was not. In the previous report, it was suggested that this might be due to the threshold on the indicated airspeed being set too high. This may, in fact, be the case, but lowering it would also tend to intensify the phenomenon in Table I of an airspeed failure being declared before the appropriate angle-of-attack failure being detected.

The inability of the method to detect INS gyro torque failures and AHRS gyro servo failures was attributed to the fact that these would cause low magnitude drifts, on the order of earth rate. No explanation was offered for the case of loss of cutoff for the directional gyro.

Failures that produced large drifts were readily detected, expecially in the case of an INS gyro float leak. A "false" alarm of INS bank was indicated for all of these test runs. However, a failure in one gyro will cause erroneous output data along other gimbal axes as well,

TABLE II

DETECTION OF SUDDEN FAILURE WITH DRIFT EFFECTS

Failure: Erroneous Output	Location	Simulation	Time Before Detection (seconds)	False Alarms
Clogged static line: Altitude, vertical velocity	Descent	$\tau_p = 100$	1.4, 2.0	--
Clogged pitot line: Indicated airspeed	Turn	$\tau_p^* = 100$	∞	--
	Descent	$\tau_p^* = 100$	∞	--
INS vertical gyro torquer failure: Pitch, bank, or both	Level	$\zeta_1^g = 1$	∞	--
INS heading gyro torquer failure: Heading	Level	$\zeta_3^g = 1$	∞	--
INS gyro float leak: INS heading	Level	$\bar{\xi}_{11}^{g\dagger} = 0.0025$	0.6	INS bank (7.2 s)
		$\bar{\xi}_{11}^g = 0.005$	0.4	INS bank (5.0 s)
		$\bar{\xi}_{11}^g = 0.01$	0.2	INS bank (3.6 s)
		$\bar{\xi}_{11}^g = 0.02$	0.2	INS bank (2.8 s)
		$\bar{\xi}_{11}^g = 0.04$	0.2	INS bank (2.0 s)
INS gyro float leak: INS heading, INS pitch	Turn	$\bar{\xi}_{11}^g = 0.0012$	1.2, 2.2	INS bank (4.0 s)
		$\bar{\xi}_{11}^g = 0.0025$	0.8, 1.6	INS bank (2.8 s)
		$\bar{\xi}_{11}^g = 0.005$	0.4, 1.2	INS bank (2.0 s)
		$\bar{\xi}_{11}^g = 0.010$	0.2, 1.0	INS bank (1.4 s) Angle of Attack (3.8 s) Vertical Velocity (8.8 s)
		$\bar{\xi}_{11}^g = 0.020$	0.2, 0.6	INS bank (0.8 s) Angle of Attack (2.6 s) Vertical Velocity (6.2 s)
Loss of cutoff for vertical gyro: AHRS bank	Turn	$a_{vg_2}^* = 100$	4.8	AHRS Pitch (9.6 s)
Loss of cutoff for directional gyro: AHRS heading	Turn	$a_{dg}^* = 100$	∞	--
Vertical gyro servo failure: AHRS bank	Level	$a_{vg_2}^* = -1$	∞	--
Directional gyro servo failure: AHRS heading	Level	$a_{dg}^* = -1$	∞	--
Rate gyro failure: Rate gyro	Turn	$\zeta_z^{rg} = -1,\ \sigma_{rg_z} = 0$	0.6	INS heading (0.4 s) INS pitch (1.2 s) INS bank (1.2 s)

† $\bar{\xi}_{11}^g = \bar{\xi}_{33}^g = \bar{\xi}_{12}^g = \bar{\xi}_{32}^g$.

and one should expect such propagation of effects due to a failure. Note that these "false" detections occurred after the appropriate failure indications, and that for large enough failures and time for errors to propagate, such errors propagate into other portions of the model reference, causing additional false alarms.

A rate gyro failure caused an erroneous INS heading failure declaration for one algorithm iteration before the proper rate gyro failure was indicated. As was appropriately discussed in the previous report, this was due to the fact that a failed rate gyro is indicated by a number of likelihood functions surpassing their threshold values. If this does not occur approximately simultaneously (i.e., both within the same algorithm iteration period), then incorrect failure declarations will result. "Appropriate adjustment of thresholds or provision for a 'yellow zone' in the detection logic" were suggested as means of alleviating this problem. The other false alarms occurred subsequent to the proper declaration, and could probably be suppressed by appropriate logic design.

The previous study also considered the computer requirements of the functional redundancy (also denoted as "internal" redundancy) failure detection logic. Table III (from this earlier report) summarizes these requirements for implementations on three representative computers, assuming that the algorithm would be iterated five times a second. For a state-of-the-art computer, only 3.18 percent of real time would be consumed by this logic. Other requirements would include:

TABLE III

COMPUTER TIME REQUIREMENTS OF THE INTERNAL REDUNDANCY METHOD

Operation	No. Required by Method	Time Requirements (μs)		
		State of the Art	NDC-1051A	SDS-920
Adds	600	1,200	3,600	9,600
Multiplies	320	3,200	7,320	10,240
Divides	30	360	1,500	6,720
Square roots	2	100	200	1,000
Trigonometry functions	10	500	1,000	5,000
Transfers and tests	500	1,000	2,000	4,000
Total time taken for one cycle		6,360	15,620	36,560
		x5	x5	x5
Time required for five cycles		31,800	78,100	182,800
Percentage of real time		3.18	7.81	18.3

(1) less than 2000 words of storage

(2) an 18-bit or longer wordlength

(3) nine input channels A/D for attitude related quantities and seven input channels A/D for translational motion related quantities

(4) simple no/no-go output channel for each of the quantities checked by the logic.

From a first-iteration cost-effectiveness analysis, it was concluded that the cost advantage of the functional redundancy method over a hardware redundancy approach would be substantial if the algorithm could be implemented through time-sharing of an existing computer. If an additional computer were required for these calculations, the cost benefit would only be marginal, but the intention is not to provide a separate dedicated computer for this purpose.

6. OBJECTIVES OF THIS INVESTIGATION

The previous study has indicated some degree of feasibility of using functional redundancy to detect and isolate control data sensor failures. A major objective of this effort has been to improve the performance capabilities of the basic concept. In other words, it is desirable to minimize both the missed alarms and the false alarms produced by the detection logic.

Means of achieving this objective have included:

(1) Formulation of a systematic means of determining appropriate parameters for the statistical description of noises and uncertainties corrupting sensor outputs, thereby attaining optimum model references;

(2) Exploration of alternative model references, and conducting trade-off analyses of performance improvement versus additional computer loading;

(3) Development of an initialization technique that can be combined with simple model references to provide overall performance comparable to that of the more complex model references;

(4) Investigation of alternate, more systematic, means of establishing maximum likelihood estimator thresholds for declaring failures;

(5) Thorough analysis of likelihood function characteristics under normal circumstances and with failed sensors, over the envelopes of possible flight regimes, to characterize the specific aspects that differentiate an "abnormal" likelihood function from a "normal" one;

(6) Utilization of such likelihood function characterizations to determine superior detection logic, such as establishing "tight" thresholds with a required number of consecutive iterations for which the threshold is surpassed before declaration of a failure, or threshold being adaptive to amount of maneuvering as indicated by commanded or actual control surface positions;

(7) Determination of the sensitivity of detection performance to system variations that are within acceptable tolerances (as especially biases);

(8) Evaluation of the ability to detect sensor failures from a signal environment generated by a real aircraft, thereby substantiating conclusions from the analysis based on the digital simulation of aircraft and sensors.

The other major objective of this effort has been to develop the failure detection algorithm and associated digital program to a point where it can be used as a systematic design tool. Its purpose would be to aid the design of an eventual implementation of a tuned and optimized software package for a particular application of functional redundancy for failure detection. To meet this objective, the digital implementation of the algorithm has been revised to provide a maximum of design flexibility. Some of these characteristics are:

(1) The functional redundancy failure detection subroutines can be driven either by a simulation of a chosen aircraft and instrumentation or recorded data with simulated failures corrupting the signals.

(2) The statistical description of sensor errors required for the Kalman filters in the algorithm can be readily altered to correspond to any specified sensor systems, and an associated program has been written to aid in evaluation of statistics if they are not available from performance data or power spectral density evaluations of desired sensor systems.

(3) The algorithm iteration frequency can be altered.

(4) The number of samples included in each likelihood function evaluation can be set by the engineer.

(5) Strengths of "pseudonoises," used to depict the uncertainty with which the model references represent the true physical interrelationships, can be optimized to yield the best possible tracking ability of those model references. Once the optimum values are evaluated, these would be incorporated into the onboard implementation.

(6) For each likelihood function involved in the detection logic, the threshold value beyond which a failure is declared can be changed through a data input to the program.

(7) Similarly, the "time to failure declaration" parameter, specifying the time (or number of iteration periods) that a threshold must consistently be surpassed before declaring a failure, can be redefined for each likelihood function by means of data input.

To facilitate interpreting the influence of the various controllable parameters, a substantial number of outputs are available from a single run of the computer program, in both printout and plot form. These include:

(1) For each Kalman filter incorporated in the design, the difference between a filter estimate and the "true" value of that corresponding variable (available only when the aircraft and instruments are simulated, not when real data tapes are used) is printed and plotted as a function of time.

(2) The above can be compared to printouts and plots of the corresponding standard deviations (one sigma values) generated through the state error covariance matrix propagated by the Kalman filter.

The pseudonoises can be adjusted until the true differences and one sigma values correspond: such that 95% of the "true difference" values are within the 2σ envelope, or 99% are within a 3σ envelope. For this purpose, plots of the "true differences" and 1σ values from a number of simulation runs will be more useful than printouts, and these are generated by the program.

(3) Printouts and plots of the individual likelihood functions utilized in the detection algorithm are generated. The plots are especially useful in discerning the salient features of the likelihood functions under normal- and failed-sensor conditions, which would be instrumental in setting threshold and time-to-failure-declaration parameters.

(4) Printouts of threshold values, time-to-failure-declaration parameters, and time and type of failure declared during a simulated or real flight are outputted.

(5) Single likelihood function terms (N of which are added to form the likelihood function) and corresponding squared residuals are presented to aid the analysis of a large magnitude likelihood function if and when it occurs.

(6) The minimum and maximum likelihood function values in the most recent N iterations, where N is adjustable, expedite the final selection of thresholds and time-to-failure-declaration values.

(7) Periodically, all pertinent simulation (or real environment) data is printed out in addition to the model reference and likelihood function performance data.

Various modes of usage of this failure detection concept were investigated. First, the types of failures more readily or appropriately detected by other means were delineated. Thus, the eventual implementation would operate in conjunction with the initialization procedures proposed in this report, BIT, reasonableness tests, deterministic detection logic, and other methods. Failures can often be detected before being completely isolated, so different means of annunciating failures were studied. Once a failure is declared and isolated, that sensor data can either be corrected (if possible) or removed from the data stream altogether, and this aspect has also been analyzed. Finally, if a sensor has failed, there may be circumstances under which testing for recertification of that sensor would be warranted, so means of performing this function were studied. The complexity of the algorithm can range from very simple to very sophisticated, and the design philosophy of building the simplest system that provides adequate performance for a particular application is applied throughout.

This report attempts to demonstrate the performance capabilities of the functional redundancy concept in detecting and isolating sensor failures. Further, it depicts the manner in which this concept would be used in conjunction with other means of detecting failures and a systematic method of reconfiguring the overall data system once failures are detected. Once the merits of the functional redundancy concept have warranted its use in a fault tolerant system, the design tool developed herein can be exploited. Thus, a viable, cost-effective failure detection concept is presented, along with a means of incorporating it into a total data system structure.

SECTION II

THEORETICAL DEVELOPMENT

1. FUNDAMENTALS OF KALMAN FILTERING AND LIKELIHOOD FUNCTIONS

The concept of functional redundancy as a means of detecting sensor failures is dependent upon the usage of functional relationships among measured quantities as the basis of a model reference. Driving such a model with certain measured values yields model-referenced estimates of other quantities, whose measured values are available from other sensors.

A substantial number of functional relationships which can be employed are in the form of linear differential equations driven by white Gaussian disturbances. In this case, the Kalman filter is the appropriate model reference to use. Essentially, a Kalman filter is a data processing algorithm that generates the maximum likelihood estimate of the state of a linear dynamic system model, conditioned on all observed data up to the time the estimate is made. The next section describes the fundamentals of a Kalman filter implemented in discrete time; i.e., sampled-data measurements are made periodically and incorporated into the filter. This is appropriate since the filter will be implemented on a digital computer, an inherently discrete-time device.

1.1 THE DISCRETE-TIME KALMAN FILTER

It will be assumed that modelling techniques have produced an adequate system description in the form of a linear difference equation, driven by a combination of known inputs and white Gaussian noise. Linear measurements are made upon the actual system variables, and these are corrupted by white Gaussian noise.

Thus, the system state is described by

$$\underline{x}(i+1) = \underline{\Phi}(i+1,i)x(i) + \underline{B}(i)\underline{u}(i) + \underline{G}(i)\underline{w}(i) \qquad (1)$$

and the measurement on the system at time instant i is

$$\underline{z}(i) = \underline{H}(i)\underline{x}(i) + \underline{v}(i) \qquad (2)$$

in which are defined the vector variables

$\underline{x}(i)$ = n - dimensional state vector at time instant i

$\underline{u}(i)$ = r - dimensional deterministic input

$\underline{w}(i)$ = s - dimensional driving noise

$\underline{z}(i)$ = m - dimensional measurement vector

$\underline{v}(i)$ = m - dimensional measurement noise

and the system matrices

$\underline{\Phi}(i+1,i)$ = n-by-n state transition matrix

$\underline{B}(i)$ = n-by-r deterministic input matrix

$\underline{G}(i)$ = n-by-s noise input matrix

$\underline{H}(i)$ = m-by-n measurement matrix

It will be assumed that $\underline{w}(i)$ and $\underline{v}(i)$ form independent zero mean white noise sequences, each having a Gaussian density with known covariance:

$$E[\underline{w}(i)] = \underline{0} \tag{3}$$

$$E[\underline{v}(i)] = \underline{0} \tag{4}$$

$$E[\underline{w}(i)\underline{w}(j)^T] = \begin{cases} \underline{Q}(i) & i = j \\ \underline{0} & i \neq j \end{cases} \tag{5}$$

$$E[\underline{v}(i)\underline{v}(j)^T] = \begin{cases} \underline{R}(i) & i = j \\ \underline{0} & i \neq j \end{cases} \tag{6}$$

$$E[\underline{w}(i)\underline{v}(j)^T] = \underline{0} \tag{7}$$

$\underline{Q}(i)$ is a positive semidefinite s-by-s matrix, and $\underline{R}(i)$ is a positive definite m-by-m matrix (all components of the measurement vector are corrupted by white noise).

The state dynamic relation, (eq. 1), is valid for all time $i \geq 0$, once an initial condition, $\underline{x}(0)$, is specified. Since this value is not precisely known, it will be modelled as a random variable with a Gaussian probability density parameterized by a mean $\hat{\underline{x}}(0)$ and a covariance $\underline{P}_0$.

For a system modelled in this manner, the Kalman filter updates the state and error covariance estimates at a measurement sample time by

$$\hat{\underline{x}}(i) = \bar{\underline{x}}(i) + \underline{K}(i)[\underline{z}(i) - \underline{H}(i)\bar{\underline{x}}(i)] \tag{8}$$

$$\underline{P}(i) = \underline{M}(i) - \underline{K}(i)\underline{H}(i)\underline{M}(i) \tag{9}$$

where

$$\underline{K}(i) = \underline{M}(i)\underline{H}^T(i)[\underline{H}(i)\underline{M}(i)\underline{H}^T(i) + \underline{R}(i)]^{-1} \tag{10}$$

The estimates $\bar{\underline{x}}(i)$ and $\hat{\underline{x}}(i)$ are, respectively, the state estimates at time instant i, before and after the measurement $\underline{z}(i)$ is incorporated; similar meaning pertains to the error covariances $\underline{M}(i)$ and $\underline{P}(i)$, respectively.

There are alternate forms of equation (9) that are theoretically equivalent but different computationally due to finite computer wordlength. One such form would be

$$P(i) = [\underline{I} - \underline{K}(i)\underline{H}(i)]\underline{M}(i)[\underline{I} - \underline{K}(i)H(i)]^T + \underline{K}(i)\underline{R}(i)\underline{K}^T(i) \tag{11}$$

Whereas (9) is often the small difference of large numbers (especially if the measurements are very accurate), (11) is the sum of small, symmetric terms that assures positive definiteness of the resulting $\underline{P}(i)$. Also, it is less sensitive to arithmetic truncation or small errors in the computed value of $\underline{K}(i)$ than other update equations. However, it requires considerably more computation, so a performance trade-off would be necessary to determine if it warrants usage. Because computer memory and time are critical, the lower triangular form of equation (9), possibly with double precision computations, will probably be employed.

To propagate the estimate to the time of the next measurement sample, the filter relations are

$$\bar{\underline{x}}(i+1) = \underline{\Phi}(i+1,i)\hat{\underline{x}}(i) + \underline{B}(i)\underline{u}(i) \quad (12)$$

$$M(i+1) = \underline{\Phi}(i+1,i)P(i)\underline{\Phi}^T(I+1,i) + \underline{G}(i)\underline{Q}(i)\underline{G}^T(i) \quad (13)$$

These recursive relationships are initiated from the assumed Gaussian density that describes the a priori knowledge of the state:

$$\hat{\underline{x}}(0) = \hat{\underline{x}}_o \quad (14)$$

$$\underline{P}(0) = \underline{P}_o \quad (15)$$

1.2 DISCRETE-TIME REPRESENTATION OF CONTINUOUS-TIME DYNAMIC SYSTEMS

The previous section assumed a system description in the form of a linear difference equation. On the other hand, the dynamic relationships to be employed are differential equations. Thus, one requires a discrete-time system model that, as seen from the periodic (sampled-data) measurements, yields equivalent system dynamics.

Let the continuous-time model of system dynamics be

$$\dot{\underline{X}}(t) = \underline{F}(t)\underline{x}(t) + \underline{B}(t)u(t) + \underline{G}(t)\underline{w}(t) \quad (16)$$

where the differential equation for the state $\underline{x}(t)$ is driven by known inputs $\underline{u}(t)$ and a Gaussian white noise $\underline{w}(t)$ (such a noise does not exist in nature but the model is often adequate). This relationship could also be modelled somewhat more precisely by a stochastic differential equation, but the above relationship will be employed. Let $\underline{w}(t)$ have mean zero and covariance $\underline{Q}(t)$, $(t - \tau)$ with $\underline{Q}(t)$ chosen to duplicate the low frequency power spectral density of the actual noise entering the system:

$$E[\underline{w}(t)] = 0 \quad (17)$$

$$E[\underline{w}(t_1)\underline{w}^T(t_2)] = \underline{Q}(t_1)\delta(t_1-t_2) \quad (18)$$

Assume that at time instant i a measurement of the form equation (2) is taken. Further assume that the input $\underline{u}(t)$ is (essentially) constant between sample times (i.e., over a single algorithm update period). Under these assumptions, the values of $\underline{\Phi}\ (i+1,i)$, $\underline{B}(i)$, and $\underline{G}(i)\underline{Q}(i)\underline{G}^T(i)$ required in equations (12) and (13) for propagating the state estimates can be found by integrating [].

$$\frac{d}{dt}\,\underline{\Phi}(t,t_1) = \underline{F}(t)\underline{\Phi}(t,t_i) \tag{19}$$

$$\frac{d}{dt}\,D(t,t_i) = \underline{B}(t) + \underline{F}(t)\underline{D}(t,t_i) \tag{20}$$

$$\frac{d}{dt}\,N(t,t_i) = \underline{F}(t)\underline{N}(t,t_i) + \underline{N}(t,t_i)\underline{F}^T(t) + \underline{G}(t)\underline{Q}(t)\underline{G}^T(t) \tag{21}$$

from the initial conditions

$$\underline{\Phi}(t_i,t_i) = \underline{I} \tag{22}$$

$$\underline{D}(t_i,t_i) = \underline{0} \tag{23}$$

$$\underline{N}(t_i,t_i) = \underline{0} \tag{24}$$

to the time of the next measurement, t_{i+1}, and then setting

$$\underline{\Phi}(i+1,i) = \underline{\Phi}(t_{i+1},t_i) \tag{25}$$

$$\underline{B}(i) = \underline{D}(t_{i+1},t_i) \tag{26}$$

$$\underline{G}(i)\underline{Q}(i)G^T(i) = \underline{N}(t_{i+1},t_i) \tag{27}$$

These relations specify the discrete-time model that duplicates the dynamic behavior of a given linear, continuous-time system observed in sampled data fashion.

For applications in which the sample period is short compared to the dynamic system's natural modes, first order approximations to the solution of these differential equations will often suffice. These approximations are, for a sample period,

$$\underline{\Phi}(i+1,i) \cong \underline{I} + \underline{F}(t_i)\Delta T \tag{28}$$

$$\underline{B}(i) \cong \underline{B}(t_i)\Delta T \tag{29}$$

$$\underline{G}(i)\underline{Q}(i)G^T(i) \cong \underline{G}(t_i)\underline{Q}(t_i)G^T(t_i)\Delta T \tag{30}$$

Such an approximation would, however, be maintained subject to the adequacy of resulting filter performance.

1.3 EXTENDED KALMAN FILTER FOR SYSTEM WITH NONLINEAR DYNAMICS

Suppose a system were described adequately by a nonlinear dynamic relationship instead of a linear one: let the system state equation (1) be replaced by

$$\underline{x}(i+1) = \underline{f}(\underline{x}(i),\underline{u}(i),\underline{w}(i)] \tag{31}$$

where $\underline{x}(i)$, $\underline{u}(i)$, and $\underline{w}(i)$ assume the same meaning as in Section II, 1.1. For the current purposes, consider a linear measurement as in equation (2).

To propagate the filter estimate to the time of the next measurement sample, equation (12) would be replaced by

$$\underline{\bar{x}}(i+1) = \underline{f}[\underline{\hat{x}}(i),\underline{u}(i),\underline{0}] \tag{32}$$

In order to propagate the covariance matrix, as in equation (13), both $\overline{\Phi}$ i+1,i) and $\underline{G}$(i) must be evaluated. These are obtained by linearizing $\underline{f}$ ($\underline{x}$, $\underline{u}$, $\underline{w}$) about the most recent values of $\hat{\underline{x}}$, $\underline{u}$, and mean value of $\underline{w}$ (the zero vector). Thus, the component in the j-th row and k-th row of these matrices would be computed as

$$\overline{\Phi}_{jk}(i+1,i) = \frac{\partial f_j(\underline{x},\underline{u},\underline{w})}{\partial x_k}\Bigg|_{\substack{\underline{x}=\hat{\underline{x}}(i)\\ \underline{u}=\underline{u}(i)\\ \underline{w}=\underline{0}}} \qquad (33)$$

$$G_{jk}(i) = \frac{\partial f_j(\underline{x},\underline{u},\underline{w})}{\partial w_k}\Bigg|_{\substack{\underline{x}=\hat{x}(i)\\ \underline{u}=\underline{u}(i)\\ \underline{w}=\underline{0}}} \qquad (34)$$

The updates at measurement times are identical to equations (8) through (11), and the initial conditions would be given by equations (14) and (15).

1.4 LIKELIHOOD FUNCTION STATISTICAL TESTING

The model reference (Kalman filter or other functional relationship model reference) provides outputs in the form of estimates of the values of certain variables in the system dynamics. These estimates are compared to measured values of the same quantities to create error signals. Some form of test is required to deduce from the characteristics of these error signals whether something is abnormal in the system, i.e, whether a failure has occurred.

Generation of a likelihood function for the time history of each of these error signals provides one means of making such a statistical test. Conceptually, the N most recent error signal values are examined to determine whether they differ significantly from a statistical description of their values, assuming no sensor failures. The number of values utilized, N, can be varied to obtain reasonable performance. More than one would be desirable to preclude failure declarations due to single error samples of large magnitude: consistently large errors indicate abnormalities, whereas individual samples of large magnitude are to be expected. Using all samples from initial time would make the likelihood function less sensitive to sensor failures as time progressed. Consequently, a "moving window" of the N most recent samples, where N might be on the order of 5 to 20, will be considered.

Let e(i) be a given error signal at time instant i. Then the conditional joint probability density function of the most recent N error values, conditioned on previous error values, would be

$$p[e(i),\ e(i-1),\ \ldots,\ e(i-m+1)|e(i-m),\ \ldots e(1)]$$

where $p[x|y]$ is the conditional probability of the variable x, conditioned on the value of y. (To be precise, a distinction should be made between parameters used to describe a density function and actual realized values, but this will not be explicit in our notation). The particular choice of this conditional density may not be entirely clear, but it is well motivated by estimation theory.

Bayes' Rule for conditional density functions states that

$$p[a,b|c] = p[a|b,c]p[b|c] \tag{35}$$

Applying Bayes' Rule to the given density function yields

$$\begin{aligned} &p[e(i),e(i-1),\ldots,e(i-m+1)|e(i-m),\ldots,e(1)] \\ &= p[e(i)|e(i-1),\ldots,e(1)]p[e(i-1),\ldots,e(i-m+1)|e(i-m),\ldots,e(1)] \end{aligned} \tag{36}$$

Bayes' Rule can then be applied to the rightmost density in equation (36) to expand the result further. Iterating on this procedure yields

$$\begin{aligned} &p[e(i),e(i-1),\ldots,e(i-m+1)|e(i-m),\ldots,e(1)] \\ &= \prod_{j=i-m+1}^{i} p[e(j)|e(j-1),\ldots e(1)] \end{aligned} \tag{37}$$

which is the product of the conditional densities of the N most recent error values, each conditioned on the previous time history of error values.

The appropriate likelihood function for this application is the natural logarithm of the conditional probability density given by equation (37):

$$L_N(i) = \ln p[e(i),\ldots e(i-m+1) \mid e(i-m),\ldots,e(1)]$$

$$= \sum_{j=i-m+1}^{i} \ln p[e(j) \mid e(j-1),\ldots,e(1)] \qquad (38)$$

If the error sequence were in fact a set of independent, zero-mean, Gaussian random variables, this expression could be written as

$$L_N(i) = \sum_{j=i-m+1}^{i} \ln \left\{ \frac{1}{(2\pi)^{1/2}\,\sigma(j)} \exp\left[-\frac{1}{2}\,\xi^2(j)/\sigma^2(j)\right] \right\} \qquad (39)$$

where $\sigma(j)$ is the estimated variance of the j-th sample and $\xi(j)$ is a dummy variable used to define the density of $e(j)$. Substituting the realized value of the N most recent $e(j)$ values into this expression yields the likelihood function evaluated for data actually observed as

$$L_N(i) = -\frac{N}{2}\ln 2\pi - \sum_{j=i-m+1}^{i} \ln\sigma(j) - \frac{1}{2}\sum_{j=i-m+1}^{i} [e^2(j)/\sigma^2(j)] \qquad (40)$$

Thus, the likelihood function could be evaluated approximately as

$$L_N(i) = L_N(i-1) - \frac{1}{2}\left[e^2(i)/\sigma^2(i)\right] + \frac{1}{2}\left[e^2(i-m)/\sigma^2(i-m)\right] \qquad (41)$$

This relationship could be used after the first N measurements had been made to initialize the likelihood function value.

It can be shown that, if the error sequence were actually the sequence of residuals from a Kalman filter whose state equations duplicated the real system environment and whose input were a scaler measurement of the form

$$z(i) = \underline{h}^T(i)\underline{x}(i) + v(i) \tag{42}$$

then the density $\rho[e(j)|e(j-1),\ldots,e(1)]$ required in equation (38) is a Gaussian density with mean $\underline{h}^T(j)\underline{\bar{x}}(j)$ and variance $[\underline{h}^T(j)\underline{M}(j)\underline{h}(j) + R(j)]$, where $\underline{\bar{x}}(j)$ and $\underline{M}(j)$ have been defined previously. Thus, the $e^2(j)$ required in equation (41) is the squared residual,

$$e^2(j) = [z(j)-\underline{h}^T(j)\underline{\bar{x}}(j)]^2 \tag{43}$$

and the $1/\sigma^2(j)$ term is equal to

$$1/\sigma^2(j) = 1[h^T(j)\underline{M}(j)\underline{h}(j) + R(j)] \tag{44}$$

This quantity is available from the Kalman filter computations, as seen from equation (10). If more than a single measurement were incorporated into the filter, the desired $1/\sigma^2(j)$ terms could be evaluated as the diagonal terms of $[\underline{H}(j)\ \underline{M}(j)\ \underline{H}^T(j) + \underline{R}(j)]^{-1}$ (thereby neglecting off-diagonal coupling).

The assumption that the filter dynamics model duplicates the true system dynamics, and thus the assumption that the residual sequence is white, zero mean, and Gaussian, is assuredly violated for any reasonably dimensioned filter. However, substantial effort will be expended to minimize this violation, thereby providing adequate performance. Simulated failures involving biases, scale factors and drifts, as well as random noise, will demonstrate how adequate the performance is.

In order to generate the likelihood functions online, the N most recent squared error signals $e^2(j)$ and estimated variances $\sigma^2(j)$ are maintained in computer storage. As time progresses, equation (41) is used to update each likelihood function at each sample time. As can be seen from either equation (40) or (41), if $e^2(j)$ becomes consistently larger than the estimated variance, then the likelihood function will become more and more negative. A negative threshold level that the likelihood function should remain (i.e., of smaller absolute magnitude) can be determined, and then a failure can be declared if the value goes beyond this threshold. By controlling the threshold level, the number (N) of samples comprising a likelihood function, and possibly the time interval over which the threshold is exceeded before declaring a failure, the number of false alarms and missed alarms can be minimized. The last control parameter, the time-before-failure-declaration, allows tighter thresholds that do not cause false alarms due to transitory threshold surpassing; this will be discussed more extensively in Section III.2.5.

Note that the error signals that are not generated by Kalman filters also require estimated variances in the likelihood function evaluation. Since dynamic propagation is not involved, these can be provided by a priori estimated values of appropriate variance magnitudes.

2. MODEL REFERENCES

This section describes the proposed functional relationships to be employed in the detection algorithm. These will be in the form of three sets of dynamic relations, which will serve to develop three Kalman filters, and an algebraic relationship for indicated airspeed.

2.1 MODEL REFERENCE RELATING INS ATTITUDES AND AFCS BODY RATES

The Automatic Flight Control System uses three rate gyros to measure pitch rate, roll rate, and yaw rate for aircraft stabilization. This rate information is functionally related to the vehicle attitude, which is measured by the Inertial Navigation System. Let x, y, and z be the aircraft angular rotation rates about longitudinal (nose), lateral (right side), and normal (underside) axes; and ψ, θ, and ϕ be yaw, pitch, and roll angles, respectively. Then, the functional relationships are

$$\frac{d\theta}{dt} = w_y \cos \phi - w_z \sin \phi \qquad (45)$$

$$\frac{d\phi}{dt} = w_x + w_y \tan\theta \sin\phi + w_z \tan\phi \cos\phi \tag{46}$$

$$\frac{d\psi}{dt} = \frac{1}{\cos\theta}\left[w_y \sin\phi + w_z \cos\phi\right] \tag{47}$$

These equations form the basis of the mathematical model to be employed in the AFCS-INS attitude Kalman filter. Equation (47) is indeterminate if the pitch, θ, is 90° (gimbal lock condition); the algorithm might be disenabled temporarily if θ reaches the close vicinity of this value. Let the Euler angles θ, ϕ, and ψ be the three state variables of the model:

$$\underline{x} = \begin{bmatrix} x_1 \\ x_2 \\ x_3 \end{bmatrix} = \begin{bmatrix} \theta \\ \phi \\ \psi \end{bmatrix} \tag{48}$$

The rate gyro outputs are then the noise-computed inputs to this dynamic system. Thus, let the rate gyro outputs be denoted as u_1, u_2, and u_3:

$$\underline{u} = \begin{bmatrix} u_1 \\ u_2 \\ u_3 \end{bmatrix} = \begin{bmatrix} w_x \\ w_y \\ w_z \end{bmatrix} + \begin{bmatrix} w_1 \\ w_2 \\ w_3 \end{bmatrix} \tag{49}$$

Thus the true rates are corrupted by the white Gaussian noise $\underline{w}$, used to model the noise and uncertainty inherent in the rate gyros. Using this notation, the dynamics to be incorporated into the Kalman filter are

$$\frac{d}{dt}\begin{bmatrix} x_1 \\ \\ x_2 \\ \\ \\ x_3 \end{bmatrix} = \begin{bmatrix} (u_2-w_2)\cos x_2 - (u_3-w_3)\sin x_2 \\ \\ (u_1-w_1) + (u_2-w_2)\tan x_1 \sin x_2 \\ + (u_3-w_3)\tan x_1 \sin x_2 \\ \\ \frac{1}{\cos x_1}[(u_2-w_2)\sin x_2 + (u_3 - w_3)\cos x_2] \end{bmatrix} \tag{50}$$

or

$$\frac{d\underline{x}(t)}{dt} = \underline{f}_c\ [\underline{x}(t),\ \underline{u}(t),\ \underline{w}(t)] \tag{51}$$

The simplest means of generating an approximately equivalent discrete-time equation to propagate the state estimate from one sample time to the next would be to use Euler integration:

$$\begin{aligned} \underline{x}(i+1) &= \underline{f}[\underline{x}(i),\ \underline{u}(i),\ \underline{w}(i)] \\ &= \underline{x}(i) + T\ \underline{f}_c[\underline{x}(i),\ \underline{u}(i),\ \underline{w}(i)] \end{aligned} \tag{52}$$

where T is the sample period for the algorithm update.

As pointed out in the previous report [], the accuracy of this approximation is improved if the value of $\underline{u}(t)$ at the midpoint of an integration interval were used instead of its value at the beginning of the interval. At time instant (i+1), $\underline{u}(i+1)$ is available as a measurement from the rate gyros, and $\underline{u}(i)$ can be retrieved from computer storage, and the average value 1/2 [u(i+1) + u(i)] generated and used in place of $\underline{u}(i)$ in equation (52).

There are more accurate methods of updating nonlinear dynamic equations, but unless this technique does not yield adequate performance, it would be best to use a simple routine that does not burden computer time or memory.

The attitude measurements from the INS are also corrupted by noise and uncertainties, and thus its outputs at a given time instant i, denoted as $\underline{z}(i)$, are modelled as

$$\underline{z}(i) = \begin{bmatrix} z_1(i) \\ z_2(i) \\ z_3(i) \end{bmatrix} = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} \theta(i) \\ \phi(i) \\ \psi(i) \end{bmatrix} + \begin{bmatrix} v_1(i) \\ v_2(i) \\ v_3(i) \end{bmatrix} \tag{53}$$

$$= \quad H \quad \underline{x}(i) + \underline{v}(i)$$

where $\underline{v}(i)$ is a zero mean, white, Gaussian noise. It is assumed to be uncorrelated with $\underline{w}(i)$, so that the statistics required for the Kalman filter are given by appropriate $\underline{R}(i)$ and $\underline{Q}(i)$ matrices, the covariances of $\underline{v}(i)$ and $\underline{w}(i)$, respectively.

Thus, the overall mathematical model to be used in formulating the Kalman filter would be as in Figure 1.

Let the actual attitude measurements from the INS be denoted as $\theta_{INS(i)}$, $\phi_{INS(i)}$, and $\psi_{INS(i)}$. Generated by the Kalman filter are predictions of what these values should be, before the measurements are actually taken; let $\bar{x}_1(i)$, $\bar{x}_2(i)$, and $\bar{x}_3(i)$ represent these values. Then the three residuals of interest are:

$$e_1(i) = \theta_{INS(i)} - \bar{x}_1(i) \tag{54a}$$

$$e_2(i) = \phi_{INS(i)} - \bar{x}_2(i) \tag{54b}$$

$$e_3(i) = \psi_{INS(i)} - \bar{x}_3(i) \tag{54c}$$

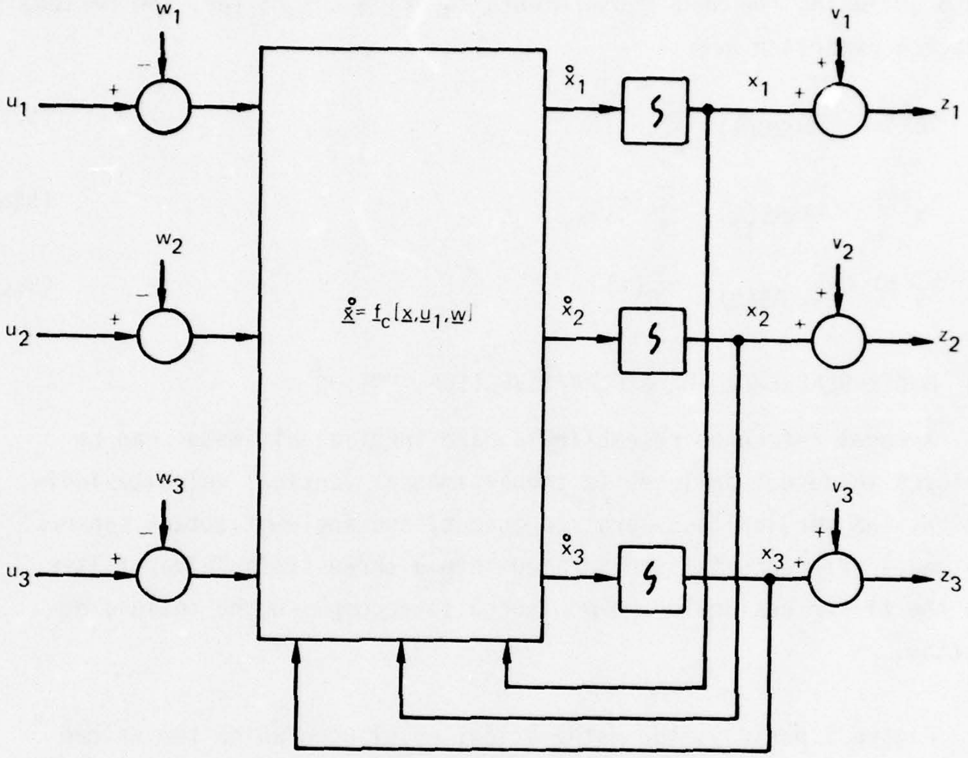


Figure 1. Schematic of Attitude Dynamics Model

2.2 MODEL REFERENCE FOR AHRS ATTITUDES AND AFCS BODY RATES

The same basic model reference is used to relate the AFCS gyro body rates and the attitude indication of the Attitude and Heading Reference System. By replacing the subscripts 1, 2, and 3 by the indices 4, 5, and 6 and using the AHRS measurements to drive the filter, the residuals that are generated are

$$e_4(i) = \theta_{AHRS(i)} - \bar{x}_4(i) \tag{55a}$$

$$e_5(i) = \phi_{AHRS(i)} - \bar{x}_5(i) \tag{55b}$$

$$e_6(i) = \psi_{AHRS(i)} - \bar{x}_6(i) \tag{55c}$$

2.3 MODEL REFERENCE FOR AIRCRAFT VERTICAL MOTION

A model reference resembling a baro-inertial altimeter can be employed to detect failures in the altimeter, vertical velocity indication, INS vertical acceleration output, and angle-of-attack sensor. The model reference is incorporated into a three-state Kalman filter, and the filter residuals are monitored to accomplish the failure detection.

Figure 2 portrays the mathematical model upon which the Kalman filter is based. The INS accelerometer output is modelled as the true specific gravity plus a white Gaussian noise (the zh axis points downward, thus causing the negative sign); the noise and the value of gravity are subtracted from that accelerometer output to yield the "true" vertical acceleration. This is integrated twice to yield altitude, which is then put through a first order lag to model the lag between the altimeter reading and the true altitude. The three state variables are identified in the figure as x_7 = lagging altitude, x_8 = true altitude, and x_9 = true vertical velocity.

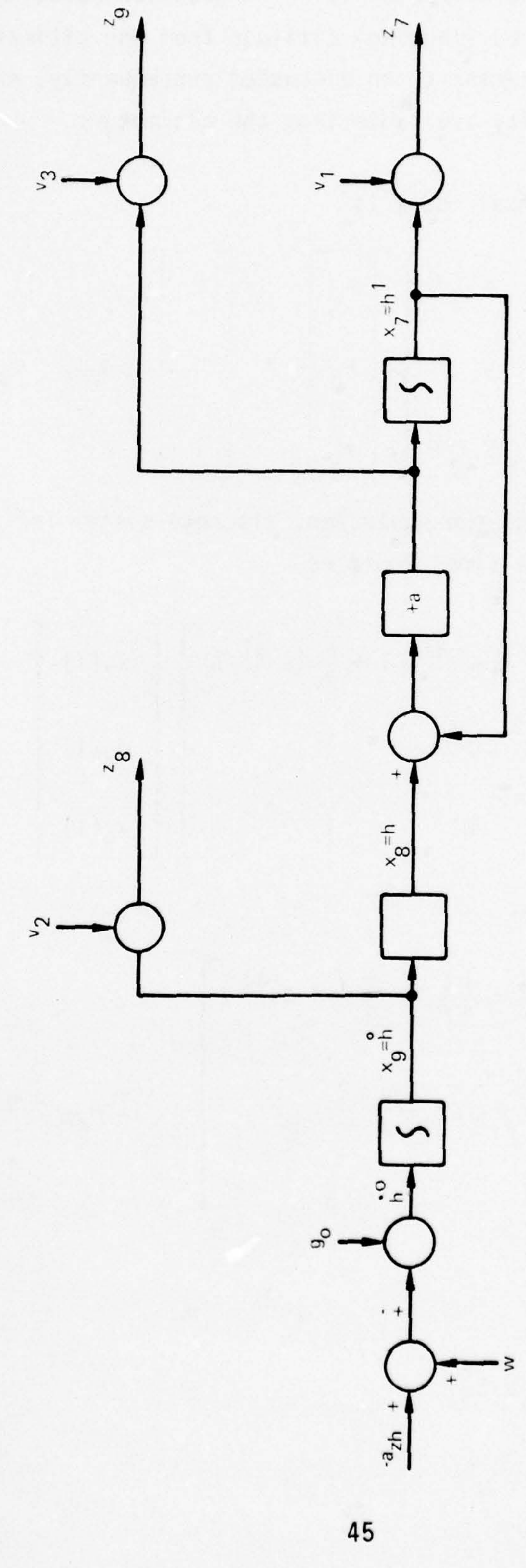


Figure 2. Vertical Motion Dynamics

The measurements are corrupted by white Gaussian noise, and are comprised of z_7 = measured (lagging) altitude from the altimeter, z_8 = vertical velocity measurement to be discussed subsequently, and z_9 = measured vertical velocity available from the altimeter.

Thus, the mathematical model is

$$\dot{\underline{x}} = \begin{bmatrix} \dot{x}_7 \\ \dot{x}_8 \\ \dot{x}_9 \end{bmatrix} = \begin{bmatrix} -a & a & 0 \\ 0 & 0 & 1 \\ 0 & 0 & 0 \end{bmatrix} \begin{bmatrix} x_7 \\ x_8 \\ x_9 \end{bmatrix} + \begin{bmatrix} 0 \\ 0 \\ 1 \end{bmatrix} [-a_{zh} - g_o + w] \quad (56)$$

A simple approximation to the equivalent discrete system for propagating estimates between update times would be

$$\begin{bmatrix} x_7(i+1) \\ x_8(i+1) \\ x_9(i+1) \end{bmatrix} = \begin{bmatrix} e^{-aT} & \left(1-e^{-aT}\right) & T + \frac{1}{a}\left(e^{-aT}-1\right) \\ 0 & 1 & T \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} x_7(i) \\ x_8(i) \\ x_9(i) \end{bmatrix} \quad (57)$$

$$+ \begin{bmatrix} T\left(\frac{T}{2} - \frac{1}{a}\right) + \frac{1}{a^2}\left(1-e^{-aT}\right) \\ 1/2\ T^2 \\ T \end{bmatrix} [-a_{zh} - g_o + w]$$

This is in the form of $\underline{x}(i+1) = \underline{\Phi}(i+1,i)\ \underline{x}(i) + \underline{B}(i)\ \underline{u}(i) + \underline{G}(i)\ \underline{w}(i)$ where $\underline{B}(i) = \underline{G}(i)$. As mentioned previously, by employing the average value of a_{zh} over an interval instead of its value at the beginning of the interval, i.e., replacing $a_{zh}(i)$ by $1/2\ [a_{zh}(i) + a_{zh}(i+1)]$ when propagating to time instant (i+1), superior integration accuracy is obtained.

Two formulations of the measurement vector are possible, one composed of z_7 and z_8 and the other including z_9 as well. The measurements z_7 and z_9 are the altitude and vertical velocity derived from the same source, barometric altitude determined from the static pressure source. Hence, a failure in this single source would invalidate both signals, and so the first formulation uses a vertical velocity measurement independent of the altitude reading, z_8. Thus, an inconsistency between two sources of information could be detected. The second formulation includes both vertical velocity indications: it could respond more rapidly to pressure source failures, but performance was not substantially different. Consequently, the simpler two-measurement case will be depicted throughout the report.

The independent vertical velocity signal is obtained by means of the equation

$$\dot{h} = v_a(\cos\alpha \sin\theta - \sin\alpha \cos\theta \cos\phi) \tag{58}$$

Measured values of pitch, θ, and roll, ϕ, are available from the INS, and values for true airspeed, v_a, and angle of attack, α, are taken from the Air Data Computer. Under most flight regimes, the sensitivity of the computed $\dot{h}$ to errors in v_a is negligible, so that a reasonableness check on v_a is sufficient to ensure confidence in its contribution to equation (58). (Logic could disenable failure declarations for regimes of high sensitivity to v_a.) The integrity of INS pitch and roll angles can be checked by the INS-AFCS Kalman filter portion of the detection algorithm. Therefore, any discrepancy between the computed $\dot{h}$ value (considered a measurement) and the model reference estimate $\bar{x}_9$ can be attributed to a faulty angle-of-attack indication.

Referring to Figure 2, the two formulations can be summarized by

$$\underline{z}(i) = \left[\begin{array}{c} z_7(i) \\ z_8(i) \\ \hdashline z_9(i) \end{array}\right] = \left[\begin{array}{ccc} 1 & 0 & 0 \\ 0 & 0 & 1 \\ \hdashline -a & a & 0 \end{array}\right] \left[\begin{array}{c} x_7(i) \\ x_8(i) \\ x_9(i) \end{array}\right] + \left[\begin{array}{c} u_7(i) \\ u_8(i) \\ u_9(i) \end{array}\right] \tag{59a}$$

$$= \underline{H} \quad \underline{x}(i) + \underline{u} \tag{59b}$$

The partitioning in this equation depicts the two possible cases, $\underline{z}^T = [z_7, z_8]$ or $\underline{z}^T = [z_7, z_8, z_9]$.

It should be noted that, whether or not the measurement z_9 is used to drive the Kalman filter, the difference between the vertical velocimeter output, $\dot{h}_{vel}$, and the model reference estimate of this value, $[a(\bar{x}_8 - \bar{x}_7)]$, can be monitored to detect failures in the vertical velocimeter itself (as distinct from a failure that would affect both the altimeter and velocimeter).

2.4 MODEL REFERENCE FOR INDICATED AIRSPEED AND NORMAL ACCELERATION

Erroneous AFCS normal (body z axis) accelerometer output and faulty indicated airspeed can be detected by means of a fourth model reference. An independent measure of normal acceleration can be obtained from the INS outputs of platform accelerations, a_{xh}, a_{yh}, and a_{zh}, and Euler angles, ψ, θ, and ϕ, using the relation

$$\begin{aligned} a_z = {} & a_{xh}(\sin\phi \sin\psi + \cos\phi \sin\theta \cos\psi) \\ & + a_{yh}(-\sin\phi \cos\psi + \cos\phi \sin\theta \sin\psi) \\ & + a_{zh} \cos\phi \cos\theta \end{aligned} \tag{60}$$

The Euler angles and a_{zh} have previously been tested for integrity by the other model reference detection logics. Although the INS horizontal accelerometers have not been explicitly verified as operating normally, the previous report [] proposed to use this functional relationship as a means of checking the AFCS normal accelerometer (such verification could be performed by other means of failure detection). Once the value of a_z is known to be valid, it can be used to compute an alternate evaluation of indicated airspeed []:

$$v_i = \sqrt{\frac{2m\ a_z}{P_0\ S(Cn+fn\alpha)}} \tag{61}$$

where m is the aircraft mass, ρ_0 is the density of air at sea level, S is the aircraft reference area, and Cn and fn are constants such that the term in parentheses is a first order approximation to the normal force coefficient, and α is the angle of attack. Note that the angle-of-attack value has also been verified previously. Comparing the result of equation (61) with the ADS indicated airspeed allows detection of failures in this measured value.

2.5 OTHER MODEL REFERENCES

Additional functional redundancies do exist in the various data systems onboard an aircraft. These were considered and rejected previously due to being

(1) infeasible or unpromising;

(2) not relevant to the task of flight stabilization and control, as being based on radiolocators or other external sources of information; or

(3) empirical relations highly dependent upon particular aircraft.

Other applications of the concept are feasible, as providing confidence in propulsion system sensors without burdening the aircraft with twenty to forty sensors per engine. However, this effort will be directed towards the control data instrumentation application to demonstrate the capabilities of the technique.

2.6 FAILURE DETECTION AND ISOLATION

Using the model references discussed in the previous sections, failures can be detected and isolated by monitoring residuals (in the case of the Kalman filters) or other appropriate error signals. Let e_1, e_2, and e_3 be the residuals of the INS-AFCS filter, as defined by equation (54). Similarly, let e_4, e_5, and e_6 be the corresponding residuals of the AHRS-AFCS filter, defined by equation (55). Further, let e_7, e_8, and e_9 denote the residuals related to the measurements z_7, z_8, and z_9 of the vertical channel filter, as described in equation (59). (Note that e_9 will be used for detection purposes whether or not it is actually used to drive the Kalman filter.) Finally, let the error between computed and measured normal acceleration define e_{10}, and the difference between computed and measured airspeed be e_{11} (see equations (60) and (61) for computation).

With these error signals defined, a particular failure can be isolated by determining which errors are growing abnormally large. Table IV depicts the isolation logic to be employed. The listing of abnormal residual magnitude pertains to initial effects. For instance, if an INS gyro fails, eventually all outputs of the INS will be affected. Note that a faulty pitch rate indication cannot be distinguished from an erroneous yaw rate measurement by this logic, but that all other failures listed can be isolated as well as detected.

TABLE IV

SENSOR FAILURES AND CORRESPONDING ABNORMAL ERROR SIGNALS

Type of Failure	Error Signal										
	e_1	e_2	e_3	e_4	e_5	e_6	e_7	e_8	e_9	e_{10}	e_{11}
INS Pitch Angle	x										
INS Roll Angle		x									
INS Yaw Angle			x								
AHRS Pitch Angle				x							
AHRS Roll Angle					x						
AHRS Yaw Angle						x					
AFCS Pitch Rate	x	x	x	x	x	x					
AFCS Roll Rate		x			x						
AFCS Yaw Rate	x	x	x	x	x	x					
ADS Altitude							x				
ADS Vertical Velocity									x		
ADS Angle of Attack								x			
INS Vertical Acceleration							x	x	x		
AFCS Normal Acceleration										x	
ADS Indicated Airspeed											x*

* = Valid only if e_{10} is not abnormally large

2.7 MODIFICATIONS TO THE MODEL REFERENCES

Previous experimental results indicated a higher sensitivity of this technique to failures in the form of excessive noise than to bias shifts. Consequently, a bias estimation capability was added to the detection algorithm by including biases as state variables in the Kalman filter model references. Actually estimating the bias levels could yield not only failure declarations due to bias shifts, but also a means of determining how to compensate such drifts to retain accurate signal levels.

Referring to Figures 1 and 2, the measurement corruption is modelled as an additive, zero mean, white Gaussian noise, as in Figure 3a. A similar diagram could be drawn for the dynamic driving noise w_i corrupting the input u_i. The model references can be altered by replacing each white noise signal with a white noise plus bias, as in Figure 3b. Note that the bias, b_i, is obtained conceptually by passing a zero mean, white Gaussian noise through an integrator. Instead of modelling a bias as

$$\dot{b}_i = 0 \tag{62}$$

as would seem to be appropriate, the model employed is

$$\dot{b}_i = w_{bi} \tag{63}$$

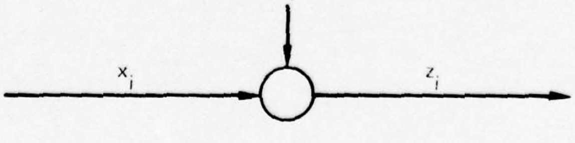


Figure 3a. White Noise Corruption; No Bias

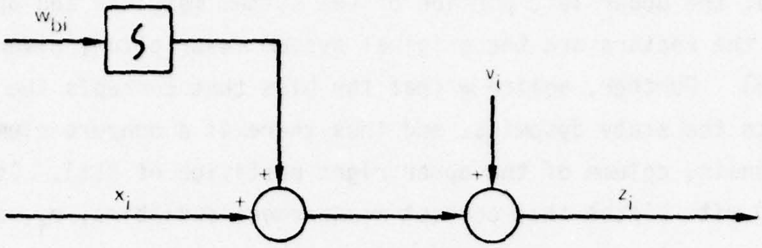


Figure 3b. Model Incorporating Bias

This "pseudonoise" is necessary to yield a Kalman filter that estimates the bias values for all time. Conceptually, using equation (62) would tell the filter mathematically that the initial value of the bias is uncertain, but you are sure the value does not change in time. As a result, the filter will use early data to estimate biases, but then essentially ignore future data (appropriately, since the filter has been "told" the values do not change in time). Putting the noise w_{bi} in says, in essence, that there is some uncertainty in the bias values for all time of interest.

First consider the model for the vertical motion dynamics, as depicted in Figure 2. Neglecting the measurement z_9, add the bias states to the input and to the two remaining measurements. The augmented system dynamics are

$$\begin{bmatrix} \dot{x}_7 \\ \dot{x}_8 \\ \dot{x}_9 \\ \hline \dot{b}_{az} \\ \dot{b}_h \\ \dot{b}_{\dot{h}} \end{bmatrix} = \left[\begin{array}{ccc|ccc} -a & a & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 \\ \hline 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \end{array}\right] \begin{bmatrix} x_7 \\ x_8 \\ x_9 \\ \hline b_{az} \\ b_h \\ b_{\dot{h}} \end{bmatrix} + \left[\begin{array}{c|ccc} 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 1 & 0 & 0 & 0 \\ \hline 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{array}\right] \begin{bmatrix} -a_{zh} - g_o + w \\ \hline w_{az} \\ w_h \\ w_{\dot{h}} \end{bmatrix} \tag{64}$$

Note that the upper left portion of the system matrices and upper portion of the vectors are the original system description, given by equation (56). Further, notice w that the bias that corrupts the input enters into the state dynamics, and thus there is a nonzero element in the corresponding column of the upper right partition of $\underline{F}(t)$. Columns associated with biases that corrupt measurement variables, z_i, are all zeroes.

The associated measurement for this system description would be

$$\begin{bmatrix} z_1 \\ z_2 \end{bmatrix} = \left[\begin{array}{ccc|ccc} 1 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 1 & 0 & 0 & 1 \end{array}\right] \begin{bmatrix} x_7 \\ x_8 \\ x_9 \\ \hline b_{az} \\ b_h \\ b_{\dot{h}} \end{bmatrix} + \begin{bmatrix} u_1 \\ u_2 \end{bmatrix} \qquad (65)$$

As before, the first partition of these quantities is the original description, with no biases, as presented in equation (59).

A Kalman filter can be developed using this dynamics model rather than the original three-state model. When adding state variables in this manner, two questions must be asked. First, is the additional complexity warranted by the performance capability gained? Secondly, is the resulting system model completely observable? In other words, can the filter see the effects of the individual states and distinguish the difference between these effects?

To answer the second question, the $\underline{F}$ matrix in equation (64) and the $\underline{H}$ in (65) can be used to generate the observability matrix,

$$\underline{\sigma} = \left[\underline{H}^T \;\middle|\; \underline{F}^T\underline{H}^T \;\middle|\; --- \;\middle|\; \underline{F}^{T(n-1)}\underline{H}^T\right] \qquad (66)$$

where n is the dimension of the state vector, 6 in this case. The system model is observable if and only if the rank of this matrix equals 0. If the rank is (n-k), then there are k unobservable state variables.

When this test is conducted, it is found that only five of the six states are observable. This is true whether z_9 is included or not, and so its inclusion would not be warranted from an observability standpoint.

If the altimeter bias state, b_h, is removed from the model, the rank of the resulting observability matrix is still five, and thus the model is completely observable. Thus, the filter can estimate the bias in the vertical accelerometer and in the vertical velocity measurement, but cannot separately identify the bias in the altimeter.

This filter formulation has been programmed and combined with the flight simulation program. Whether the added complexity yields substantial enough performance improvement to merit implementation will be discussed subsequently. However, an attractive alternative to bias estimation will also be proposed, improving performance but not increasing the state vector dimension.

A similar state augmentation technique for bias estimation can be applied to the AFSC-INS and AFCS-AHRS Kalman filters as well. Corresponding to equation (51) would be the augmented equation

$$\frac{d}{dt}\begin{bmatrix} \underline{x}(t) \\ b(t) \end{bmatrix} = \begin{bmatrix} \underline{f}_c[\underline{x}(t),\underline{b}(t),\underline{u}(t),\underline{w}(t)] \\ \underline{w}_b(t) \end{bmatrix} \tag{67}$$

and replacing (53) would be

$$\underline{z}(i) = \left[\, \underline{H} \;\vdots\; \underline{H}_b \,\right] \begin{bmatrix} \underline{x}(i) \\ \underline{b}(i) \end{bmatrix} + \underline{v}(i) \tag{68}$$

Because scale factor errors appeared to be considerably more significant than bias errors in rate gyros [ ], additional bias states were not added to the rate gyros in the model reference. (Such states could readily be included however.) Instead, bias states were added only to the Euler angles in the model reference, corresponding to the outputs of either the INS or AHRS. Thus, $\underline{b}$ is a three-dimensional vector, and $\underline{f}_c$ is not affected by $\underline{b}$ but is the original $\underline{f}_c$ in equation (51). To specify the associated Kalman filter completely, a statistical description of the initial value of $\underline{b}(0)$, and the driving noise sequence, $\underline{w}_b(t)$, is required. The bias $\underline{b}(0)$ would be assumed to be of mean zero, probably uncorrelated with $\underline{x}(0)$ (though not necessarily), and Gaussian with known covariance; similarly $\underline{w}_b(t)$ would be white, Gaussian, zero mean, and uncorrelated with other random processes affecting the system.

These augmented filters have also been programmed, but are similarly regarded as means of improving performance only if the alternative to bias estimation is inadequate. Some observability difficulty would be expected with regard to the bias added to the yaw measurement, since the yaw state, ψ, does not appear explicitly in $\underline{f}_c$ of equations (50) and (51), and might be difficult to distinguish from an assumed bias in its value. Being a nonlinear set of equations, this system description cannot readily be examined for observability as done for the vertical channel filter. However, by investigating the linearized perturbation equations corresponding to equation (67), and assuming time invariance over a period of interest, such unobservability of the bias on ψ does result. This does not demonstrate unobservability in the time-varying, nonlinear system, but does indicate a potential source of difficulty. Thus, the augmented state vector may not include the ψ bias state.

2.8 ADAPTATION TO FAILURES

In event of a sensor failure, it would be desirable to synthesize a best estimate of the parameters whose direct measurements have been lost. Such adaptation is feasible by inhibiting the failed signals from driving the model references. In the case of the Kalman filter references, the row of $\underline{H}$ that corresponds to the failed sensor could be set to zero and the residual not used to drive the filter. Or, the corresponding term in the covariance matrix $\underline{R}$ could be increased appropriately to de-emphasize the value of a sensor reading if a "hard" failure has not occurred and there is still some limited information in the signal.

It would be conceivable to use such a technique to synthesize the values of

(1) any INS Euler angle

(2) any AHRS Euler angle

(3) ADS altitude, vertical velocity, or indicated airspeed

(4) AFCS normal acceleration.

It is important to know whether all states in the various Kalman filters are observable in the event of a failed sensor being removed from the data inputs. The rank of associated observability matrices will indicate such capability.

First consider either of the two attitude Kalman filters. As before, looking at the linearized systems indicates that a viable estimate of the yaw angle may be difficult to obtain if the INS or AHRS yaw signal fails. Perhaps the best procedure in such a case would be to use the other available yaw signal (or signals if hardware redundancy is also employed) since the filter cannot provide useful information. However, if either the pitch or roll (or both) indications are lost, a somewhat degraded filter estimate of all states is still attainable.

To examine the extent of performance degradation due to removal of failed signals, it may be useful to look at the (steady-state) value of the information matrix, i.e., the matrix $\underline{P}^{-1}$ that is propagated from $\underline{P}^{-1}(t_0) = \underline{0}$ [ ]. This could be accomplished by setting the appropriate term in $\underline{R}$ to infinity or, equivalently, the corresponding term in $\underline{R}^{-1}$ to zero, and directly computing $\underline{P}^{-1}$ in the limit from an initial condition. This technique is probably better suited to the case of linear dynamics, as in the vertical channel filter to follow.

Now consider the vertical channel filter driven by z_7 and z_8. If z_8, computed vertical velocity, were removed, all states are still observable, and thus a viable estimate can be maintained. However, if z_7, altitude, were lost, the observability matrix is of rank one, and only vertical velocity is observable.

If z_9, the vertical velocity derived from barometric altitude, were also used to drive the vertical channel filter, then loss of z_8 would again yield a completely observable system, as would loss of both z_8 and z_9. If z_7 or both z_7 and z_8 were lost, two states are observable due to the measurement z_8.

3. OPTIMAL COMBINATION OF DATA

Conceptually, the Kalman filters employed as model references can serve to generate optimum estimates of the model state variables. Thus, the AFCS-INS or AFCS-AHRS filters could provide optimum estimates of the Euler angles. In fact, an overall "optimum" estimate of Euler angles could be generated by a larger filter that incorporated data from all three systems: the AFCS rate gyros, the inertial system, and the attitude and heading reference system. Similarly, the vertical channel filter could conceptually provide optimum estimates of altitude, lagging altitude, and vertical velocity. By combining the information from individual data systems, an optimal estimator can increase the precision of the data above that of any single system. Consequently, one might propose to use the outputs of the Kalman filters as the best signals to represent these variables.

However, the simplicity of the models employed in the filters dictates against this. A truly optimum filter, incorporating as accurate (and complex) a model of a certain dynamic phenomenon as can be developed, will in fact yield estimates whose precision is higher than any single data source. The design objective here has not been to develop a large optimal state estimator, but to generate as simple and small an estimator as will provide adequate failure detection performance. Not only are the filter dimensions low, but in the case of the two attitude filters, the nonlinear dynamic equations are propagated by a simple first order integration technique at a rather low update rate.

Although the prospects of the filters serving as adequate data estimators seemed poor, tests were conducted to determine realizable performance. Both the separated AFCS-INS and AFCS-AHRS attitude filters and the combined AFCS-INS-AHRS filter were tested by means of simulated aircraft and measurement system dynamics, as was the vertical channel filter. The attitude results were poor, especially during any substantial maneuvering, this being attributed mostly to the simplified propagation of nonlinear dynamics. In fact, the problem of transient filter response manifests itself to some degree in the failure detection logic, but the effects can be masked by procedures to be described in the next section. As a result of these procedures, the simple filters can serve for the failure detection function, but the filtered estimates themselves are too inaccurate to use as optimal data signals. The vertical channel filter exhibited better performance, but any avionics system involving inertial and air data systems will encompass baro-inertial coupling to damp the inertial vertical channel, so there is no significant gain from using this filter to combine data from individual sensors.

It is conceivable that a data system that does in fact perform optimal combination of information from the INS, AHRS, AFCS, and ADS will be developed. Such a system would require more accurate models, accounting for biases and other phenomena not modelled herein. Update rates and propagation (integration) techniques would require further investigation. Practical aspects, such as the fact that the various sensors are situated at different environments (as vibration effects), would also have to be considered. If such an overall data system were designed into a vehicle's avionics system, it would be ideally suited to exploiting the concept of functional redundancy. However, this effort has been conducted without imposing the assumption that such an avionics architecture were available: demonstration of concept feasibility with a minimum of extra onboard computer loading has been a very influential design objective.

SECTION III

PRACTICAL APPLICATION OF THE TECHNIQUE

1. SYSTEM DESIGN AND IMPLEMENTATION

The feasibility of using functional redundancy to detect and isolate control data sensor failures has been partially established in the past [ ]. Numerous means of improving the performance capabilities of the concept have been developed in this research, with substantial success in minimizing the missed alarms and false alarms produced by the detection logic. However, a mere demonstration of concept feasibility is not as desirable or useful as such a demonstration combined with a methodical, systematic procedure of application of the concept. This chapter describes two principal aspects of applying the functional redundancy method of failure detection to practical situations.

First of all, the development of the failure detection algorithm and associated digital computer software into a systematic design tool will be delineated. In so doing, the various methods used to enhance the algorithm performance capabilities will be thoroughly discussed. The result of these improvements is a software package with sufficient flexibility to allow an engineer to tailor the failure detection algorithm to his particular needs. Once the design has been optimized with this tool, final implementation of the software in an onboard computer can be conducted.

This is the second aspect of practical application: once a tuned design has been achieved, what mode of usage is most appropriate? As conceived herein, functional redundancy will be used in conjunction with other failure detection methods. Within this framework, there are many alternative ways of declaring failures, and once a failure is declared, there are numerous methods of restructuring the data systems. The performance analysis provided by the design tool can aid the selection of the most advantageous option for a particular application. Whether it be simple or sophisticated, the end result will be a means of (1) detecting, (2) isolating, and (3) declaring failures, combined with a logic for (4) reconfiguring the data system, that is effective and efficient for on-line use.

2. USE OF DESIGN TOOL

The computer software that has been developed is in four basic parts. First an all-digital aircraft flight simulator generates the actual profiles to be flown. It is a complete and sophisticated simulation program, encompassing not only basic flight path equations, aircraft translational dynamics and attitude relations, but also detailed models of atmospheric effects, winds, the vehicle's engines, aerodynamic effects, and the flight control system employed (including its influence in generating sideslip phenomena). The extensive detail of this simulation program provides a very accurate representation of true flight characteristics. The output of this segment of software is the set of "true" values of parameters to describe the aircraft operation and the environment in which it is flying.

These outputs then feed into the second general partition of the software, the models of the instrumentation systems onboard the aircraft. Included are segments that completely define the operational characteristics of:

(1) the Air Data System (ADS),
(2) the Inertial Navigation System (INS),
(3) the Attitude Heading Reference System (AHRS), and
(4) the Automatic Flight Control System (AFCS) data sensors.

These instrumentation models include sensor dynamic characteristics and sources of uncertainty. In all cases, the parameters that define the instrument operation, power spectral densities of noises and/or uncertainties inherent in the instruments, and signal biases can be readily altered by means of input cards to the program. Thus, the first two segments of the program allow the specification of any aircraft in any environment with any complement of particular data sensor systems. Moreover, off-nominal as well as nominal situations can be simulated, as an F-4 with a different, more state-of-the-art INS than these aircraft actually carry (as was actually done in the particular performance analyses reported herein).

The instrumentation model segment of the software performs another function as well. By proper selection of input cards to the program, the engineer can cause this program segment to simulate a wide variety of instrumentation failures. The failures that are simulated duplicate the major modes of failure described in Section I.4.

The third section of the software package consists of a very flexible set of logic for detecting, isolating, and declaring failures and for restructuring the data system upon failure declaration. In actual onboard implementation of the functional redundancy concept, the software would be simpler: the flexibility is intended to expedite initial design procedures. Two versions of this segment have been programmed--that encompassing the "standard" Kalman filter structures and the other that employs the augmented filters for bias estimation as well; the separation into two interchangeable segments rather than one large program with options was motivated by computer programming efficiency.

The final section provides performance evaluation outputs in the form of both printouts and plots of significant parameters. By monitoring these outputs, the engineer can iterate upon a failure detection logic design until he converges upon a final implementation with a performance suited to his needs.

If desired, the computer software is readily modified to accommodate actual flight data recorded from the appropriate sensors onboard an aircraft, rather than being driven by the simulation. Sampled data from the tapes of the sensor outputs would be read into computer locations from which the third and fourth software segments, the detection logic and performance evaluation segments, are driven. (A portion of the performance evaluation segment is inhibited since the "true" values of flight variables are not separable from the data - this will be developed further in paragraph 2.3 of this section. Sensor failures can still be simulated by generating the sensor output signal variations due to a failure (generated by two runs of the overall simulation program, one without failures and the second identical to the first, but with a failure simulated - the sensor outputs are then differenced to obtain a time history of the desired signal variation) and adding this to the real data samples.

The following section will discuss the methodical design procedure made available by this design tool, along with associated concepts and software.

2.1 BASIS OF COMPARISON

The initial computer runs are conducted with no simulated failures and sensor biases set to zero. For the current investigation, a nominal trajectory was chosen to be a simulated approach trajectory flown by an F-4, composed of a period of level flight followed by a coordinated final turn and then a pitchover and descent to touchdown. This choice was made to compare performance results to those of the previous investigation, and there is nothing inherent in the software to constrain attention to only this trajectory.

There are a number of reasons for such a set of computer runs. First of all, the entire nominal trajectory is flown and appropriate data is stored to provide realistic values for aircraft and logic parameters at various selected points along the trajectory. These can then serve to initialize the simulation of shorter trajectory segments, on the order of 10 to 30 seconds of flight time, during which failures or other phenomena can be simulated. In the tests conducted in this investigation, three such segments were chosen: one in level flight, one for the duration of the turn, and the last during descent. Thus, the particular types of flight environment deemed to be critical to performance evaluations can be simulated realistically with only small amounts of required computer time.

Another reason for a set of trajectories with no simulated failures or sensor biases is to allow "tuning" of the filters embodied in the detection logic. Means of obtaining good statistical data about sensor performance characteristics will be discussed in paragraph 2.2 which follows. Such information is required to establish the covariance matrices $\underline{Q}$, $\underline{R}$, and $\underline{P}_0$ that define the Kalman filters. However, even with good statistical data about the sensors, establishing appropriate covariances is an iterative process. Consider either of the two attitude filters: the $\underline{Q}$ matrix embodies not only the uncertainty in the rate gyro outputs, but also the uncertainty contributed by using a very simple mathematical model to represent a complex dynamical relationship. Consequently, it is necessary to vary these covariance matrices until desirable filter performance is obtained.

In practice, this tuning is achieved by making repeated runs of a nominal trajectory while changing only the filter covariances from one run to the next. Then the filter's evaluations of the standard deviations in its own state estimates are obtained by taking the square root of each diagonal term of the propagated error covariance $\underline{P}$. These are then compared to the observed time history of the components $(\hat{\underline{x}} - \underline{x}_t)$, where $\hat{\underline{x}}$ is the filter estimate of the state and $\underline{x}_t$ is the "true" value of the state variables as generated by the simulation portion of the software. If approximately 70% of the time history of each separate component of $(\hat{\underline{x}} - \underline{x}_t)$ is within the propagated standard deviation value from zero, or if about 95% are within two times this value, then the filter is fairly well "tuned." Typically, the elements of $\underline{Q}$ especially have to be increased over sensor statistics magnitudes to preclude a substantial underestimate of error standard deviations by the filters. Paragraph 2.3 of this section will describe the capabilities of the current software to facilitate this timing.

Sensitivity of this tuning to sensor biases that are within tolerances may also be considered during this tuning. As a result, the magnitudes of the noise covariances may be increased. Or, the tuning based on zero biases might be maintained and the thresholds in the detection logic adjusted to accommodate the in-tolerance bias effects.

These initial data runs generate plots of time histories of each individual likelihood function used in the failure detection logic. Thus, their character under normal conditions can be investigated. By simulating all of the pertinent aircraft flight profile and in-tolerance system variations, a complete analysis of likelihood function maximum magnitudes and transient characteristics under normal operation can be attained. This serves as one basis of setting the thresholds and time-before-failure-declaration parameters, to be discussed subsequently.

2.2 ESTABLISHING SENSOR STATISTICS

Obtaining statistical information about the error characteristics of sensors from their manufacturers or users is, in general, very difficult. Consequently, this information, which is required not only to design the filters of the failure detection algorithm but also to generate a realistic simulation for performance analyses, often must be generated by the system designer. There are standard techniques available that facilitate the evaluation of reasonable variance values for noise and uncertainty phenomena that corrupt sensor outputs. For instance, power spectral density analysis of signals can be used to verify the form error models in the simulation as well as determine appropriate noise levels to drive the models. For the simplified models in the Kalman filters of a true value being corrupted by a white Gaussian noise, the strength of the noise can be set so as to duplicate the low frequency power spectral density value.

A data reduction program has been developed to perform a statistical analysis of a sequence of data samples, consisting of evaluations of the mean and variance of a set of samples and a test for the whiteness of the sequence. There are three primary applications for it with regard to functional redundancy logic design and implementation:

(1) A sensor, or a number of identical sensors, can be tested under controlled conditions so that the true value of the variable being measured is known. Then, based on the assumption inherent in the detection logic filters of the instrument being adequately modelled as

generating the true value corrupted by white Gaussian noise (and possibly a constant bias that can be subtracted out), a valid variance level for that corrupting noise can be established.

(2) Again under controlled (laboratory) conditions, the more complex simulation models can be validated and good model parameters attained. Conceptually, an (extended) Kalman filter would be developed about a given dynamics model of each measuring device, real sensor data would be used to drive the filter, and the mean, variance, and whiteness of the resulting residual sequence tested. Iterations of this hypothesis testing would yield the final simulation model specification.

(3) Another application will be discussed further in paragraph 2.8 of this section, namely that of preflight initialization. Again, under conditions that allow true values of measured variables to be known exactly, the sensor systems would be operated and the appropriate values of $\underline{R}$, $\underline{Q}$, and $\underline{P}_0$ could be established before each operational usage of the filters. This would allow adaptation to component variations. Moreover, by estimating the mean value of a signal whose appropriate value is known (since the true variable value that the signal represents is known), the bias in the signal can be estimated and compensated.

The data reduction program operates in the following manner. First one establishes a known steady-state value of what the particular instrument should be measuring, and runs the sensor in this steady-state condition (or possibly lets the variable assume a known nominal function of time). This data is assumed to be in sampled data form, using a fixed sample rate.

Once this data is generated, an even integer N is chosen as the number of samples over which the mean and variance of the sequence can be assumed to remain essentially constant. A sliding arc of N samples at a time is then used to estimate the mean and variance evaluated at the time of the middle sample in the sliding arc. In other words, to determine the mean and variance values for time instant i, the data samples from instant (i - 1/2N) through instant (i + 1/2N - 1) would be used, a total of N samples at a time. This N-sample arc is allowed to "slide" one sample period at a time, generating a sequence of mean and variance values. No evaluations are made for i such that (i - 1/2 N) 1 or (i + 1/2N - 1) (total number of data samples). Thus, for data samples x(1), x(2), ..., the mean evaluated for instant i, denoted as m(i), is

$$m(i) = \frac{1}{m} \sum_{j=i-\frac{1}{2}m}^{i=\frac{1}{2}m-1} x(j) \qquad (69)$$

$$= m(i-1) + \frac{1}{N} [x(i+\tfrac{1}{2}m-1)-x(i-\tfrac{1}{2}m-1)] \qquad (70)$$

and the associated variance, v(i), would be calculated as

$$v(i) = \frac{1}{N-1} \sum_{j=i-\frac{1}{2}m}^{i+\frac{1}{2}m-1} [x(j) - m(i)]^2 \qquad (71)$$

If the appropriate "flag" parameter is set in the data reduction program input, it will also perform a Q-test to determine whether the sequence of data samples is a white sequence or not. Such information is useful in verifying the adequacy of assumed models. The manner in which the Q-test indicates the whiteness of a sequence, or the degree to which its consecutive values are not correlated with one another, is described in Reference [ ].

2.3 SOFTWARE INPUTS AND OUTPUTS

The applicability of the software package as a design tool is a function of the flexibility provided in both input controls and available performance analysis outputs. First the significant inputs will be considered, including control over the inputs to the error detection logic:

(1) aircraft and trajectory simulation

(2) sensor error model parameters

(3) random number generators

(4) failure simulations

(5) replacement of simulated data with real data recorded in flight test, and control over the detection logic itself

(6) dimension of filters employed (inclusion or exclusion of bias estimation)

(7) the statistical description of sensor errors embodied in the filters

(8) the strengths of "pseudonoises" added to the Kalman filter system models to depict the uncertainty in the accuracy of the models themselves

(9) the number of samples included in each likelihood function evaluation

(10) the threshold for each likelihood function in the failure detection logic

(11) the "time-to-failure-declaration" associated with each likelihood function in the detection logic

(12) the algorithm iteration rate.

The first five items (except for the third) have been discussed previously in Section III.2, and are reiterated here to emphasize the ease of altering these control inputs. Any aircraft configuration, with any complement of particular sensor systems, can be flown on any specified trajectory through various environments. Data collected from these first computer runs serve to initialize the simulation at various points of interest along the trajectory. Shorter trajectory segments are then flown from these points, with any of an array of failures and in-tolerance system variations simulated during the shorter segments of flight profile (the specification and length of which are also under complete control of the designer).

The third item listed requires further elaboration. The simulation (or real data driving inputs) entail a specification of detection logic performance for a single set of sensor data. It is not a covariance type analysis in which a statistical description of expected performance over an ensemble of flights is generated in a single computer run. Rather, because nonlinearities in simulation models preclude such an analysis, Monte Carlo runs must be generated in order to assume a statistically significant specification of system performance. Uncertainties and noise phenomena are simulated by means of random number generators and appropriate weighting to generate white Gaussian sequences of values. By controlling the initial value from which the random number generators start, different sequences of values are generated so as to share identical statistics, thereby allowing Monte Carlo runs of the same nominal situation to be made. The software has been written so that, unless otherwise specified, the initial value in the noise generators is always the same for the start of any data run; this is to allow comparison of performance over simulations which are known to be exactly the same except for some controlled parameter, as the incorporation of exclusion of a sensor failure. However, by making the multiple passes over the same trajectory in a single data run, the random number generators are controlled so that a Monte Carlo set of runs is in fact generated.

The first two facets of detection logic control have already been discussed. As mentioned previously, two separate software packages have been developed, one with bias estimation and the other without. For any particular application, trade-off analyses of performance improvement versus additional computer loading caused by bias estimation should probably be conducted. However, due to factors to be discussed further in Section III.2.8, the simpler version will most likely be preferable. Unless otherwise noted, this report will be portraying the performance of this version.

Section III.2.2 discussed some methods of developing a good statistical description of the sensors that drive the detection logic filters in an actual implementation. This would be the first step in setting the values of $\underline{Q}$ and $\underline{R}$ in these filters.

However, such evaluations of $\underline{Q}$ and $\underline{R}$ are generally underestimates of values that will provide the best filter performance. This is true because the assumed models in the filters are extremely simple, and some account for the misrepresentation by these models of true sensor performance must be made. Consequently, "pseudonoises" are added to the Kalman filter models to express this uncertainty. These "pseudonoises" are typically added to the models at the same locations as the "noises" $\underline{w}$ and $\underline{v}$ enter, so that the essential result is to alter the entries of $\underline{Q}$ and $\underline{R}$ matrices. Thus, if $\underline{Q}_{SENSOR}$ and $\underline{R}_{SENSOR}$ depict the noise covariance generated to describe the sensor statistics, the actual $\underline{Q}$ and $\underline{R}$ to be employed in the filter are

$$\underline{Q} = \underline{Q}_{SENSOR} + \underline{Q}_{ADJ} \tag{72}$$

$$\underline{R} = \underline{R}_{SENSOR} + \underline{R}_{ADJ} \tag{73}$$

Here $\underline{Q}_{ADJ}$ and $\underline{R}_{ADJ}$ are matrices that are adjusted to enable the filters to achieve good $\underline{P}$ values (this determination is aided by the software outputs) and thus track adequately. Repeated runs are made while altering $\underline{Q}_{ADJ}$ and $\underline{R}_{ADJ}$ until good filter performance is attained. The software maintains a separation of $\underline{Q}_{SENSOR}$ and $\underline{R}_{SENSOR}$ from the total $\underline{Q}$ and $\underline{R}$ for convenience. By so doing, the best estimates of sensor statistics are available for reference, and the additional adjustment required due to model uncertainty can be explicitly displayed and compared to the sensor statistics.

Section II.1.4 described the application of likelihood function statistical testing to the detection of sensor failures. It was shown that the appropriate likelihood functions for the detection logic are generated approximately as an N-step sum of terms of the form $\{-1/2[e^2(i)/\sigma^2(i)]\}$ where $e(i)$ is the observed filter residual at time instant i corresponding to the variable of interest, and $\sigma^2(i)$ is the filter's estimate of what the variance of this residual error should be if there are no sensor failures. (Thus, the ability of the filters to achieve good $\underline{P}$ values will be instrumental in achieving viable detection logic performance as well as good filter tracking performance.) In other words, the N most recent residual error signal values are used to statistically test the hypothesis that no failures have occurred. If the errors are consistently larger than anticipated under the no failure hypothesis, then the likelihood function magnitude will grow abnormally large.

The value of N is a design variable. Very small values are avoided since individual error samples of large magnitude are expected even under normal conditions. On the other hand, very large values should be avoided because sensitivity to actual sensor failures would then be reduced substantially. Furthermore, the N most recent samples of data must be maintained in storage, so large N is avoided from a consideration of memory and computational loading of the onboard computer. This investigation has demonstrated that a choice of N between 5 and 20 yields good performance.

Initially, failures were declared when the likelihood function magnitude surpassed a threshold that represented the largest magnitude attained under any normal operational condition. This threshold value for each likelihood function can be altered by data input to the software package. However, analysis of the results indicated that this procedure resulted in rather high threshold magnitudes. Certain types of maneuvers would generate large transient magnitudes with no failures simulated, especially in the attitude filters. Using these magnitudes to set threshold values inhibited failure detection during straight and level flight, the type of flight regime that composed the majority of time spent in the air. Consequently, it is useful to specify both a threshold value and a parameter to indicate the time (or number of algorithm iteration periods) that the threshold must be consistently surpassed before a failure is declared. Such a "time-to-failure-declaration" parameter is also a control variable set by control data input for each likelihood function individually. This will be discussed further in Section III.2.5.

The algorithm iteration rate, or data sample rate, is also a design parameter. In this investigation, a sample period of 0.2 seconds was found to yield adequate performance without overburdening the computer capabilities in any way. Longer sample periods tended to have unacceptably long mean times to detection of failures and poor state estimate propagation in the filters between sample times, especially in the case of the attitude filters using a first order integration of nonlinear equations. On the other extreme, shorter sample periods tended to yield superior performance but the advantage gained was questionable compared to the additional computer loading.

The outputs of the software package contribute significantly to its potential use as a design tool. A single run of the program can generate a substantial amount of printout and plot data (using control input cards to determine how much is actually provided), including:

(1) For each state variable in the Kalman filters, the value of $(\hat{x} - x_t)$

(2) The corresponding error standard deviations as estimated by the filters

(3) The values of $\underline{\hat{x}}$ and $\underline{P}$, and $\underline{\bar{x}}$ and $\underline{M}$, as well as $\underline{z}$ and $\underline{u}$ for each Kalman filter

(4) For the attitude filters, the optimal estimate $\underline{\hat{x}}$ obtained by combining the two individual filters, the corresponding values of $(\underline{\hat{x}} - \underline{x}_t)$ and $\underline{P}$, and $\underline{z}_{AVG}$ obtained by combining only INS and AHRS data

(5) Individual likelihood functions

(6) Threshold values and time-to-failure-declaration parameters for each likelihood function

(7) Time and type of failures declared during the run

(8) Single likelihood function terms and corresponding squared residuals and estimated residual error covariances

(9) The minimum and maximum values attained by each likelihood function in the most recent N iterations

(10) Periodically, all pertinent simulation parameters or real environment data.

The first two outputs facilitate the setting of $\underline{Q}_{ADJ}$ and $\underline{R}_{ADJ}$ of equations (72) and (73). For a given state variable x, $\hat{x}$ is the filter estimate of its value and x_t is the "true" value as provided by the simulation. (Note again that x_t is not available when real data tapes supplant the simulations of aircraft and sensors.) The difference $(\hat{x} - x_t)$ is then printed out every iteration, and a plot of its values over the entire test trajectory is generated as well. This can then be compared to printouts and plots of the corresponding standard deviations (1σ values) as estimated by the Kalman filters. In fact, these are simply the square roots of the diagonal terms of the propagated error covariance matrix, $\underline{P}$. To "tune" the filters, the pseudonoise strengths are adjusted until the $(\hat{x} - x_t)$ sequence and the standard deviations correspond such that 95% of the true error sequence lies within the 2σ

envelope generated by the standard deviation. In practice, this tuning is accomplished more easily with plots of $(\hat{x} - x_t)$ and the 2σ envelope than with digital printout, so such plots are produced by the software.

The third set of outputs allows an evaluation of the filter state estimation capability. Both the estimates just before and just after incorporation of a measurement are included to portray the separate effects of time propagation of the state estimate and updates at measurement times. In the case of the attitude filters, this is especially valuable for determining the adequacy of the simple integration algorithm for propagation: if $\underline{\bar{x}}$ is consistently poor and $\underline{\hat{x}}$ substantially better, then some alteration of the filter propagation technique is advisable, whether it be a higher order integration technique, or the simple method applied iteratively to partitions of the time interval between measurements, or a smaller overall sample period.

With regard to the fourth set of outputs, each of the two attitude filters generates optimal estimates of the Euler angles. Denote the output of the filter driven by AFCS rate gyros and the INS as $\underline{x}_{INS}$ and $\underline{P}_{INS}$, and similarly let the outputs of the filter driven by the AFCS rate gyros and AHRS be $\underline{\hat{x}}_{AHRS}$ and $\underline{P}_{AHRS}$. If reasonable state estimation performance were achieved, it would be valuable to calculate an "over-all-optimal" state estimate, $\underline{\hat{x}}_{OVERALL}$, that combined the data from all three sensor systems. Its value would be computed as

$$\underline{\hat{x}}_{OVERALL} = \left[\underline{P}_{INS}^{-1} - \underline{P}_{AHRS}^{-1}\right]^{-1} \left[\underline{P}_{INS}^{-1}\, \underline{x}_{INS} + \underline{P}_{AHRS}^{-1}\, \underline{\hat{x}}_{AHRS}\right] \quad (74)$$

This could be approximated (with substantial decrease in computer loading) on a component-by-component basis as

$$\hat{x}_{OVERALL_i} = \frac{1}{\frac{1}{P_{INS_{ii}}} + \frac{1}{P_{AHRS_{ii}}}} \left[\frac{\hat{x}_{INS_i}}{P_{INS_{ii}}} + \frac{\hat{x}_{AHRS_i}}{P_{AHRS_{ii}}} \right]$$

$$= \frac{1}{P_{INS_{ii}} + P_{AHRS_{ii}}} \left[P_{AHRS_{ii}} \, \hat{x}_{INS_i} + P_{INS_{ii}} \, \hat{x}_{AHRS_i} \right] \qquad (75)$$

The overall error convariance would be calculated as

$$\underline{P}_{OVERALL} = \left[\underline{P}_{INS}^{-1} + \underline{P}_{AHRS}^{-1} \right]^{-1} \qquad (76)$$

Comparing the 1σ values from this $\underline{P}_{OVERALL}$ and the sequence of $(\hat{\underline{x}}_{OVERALL} - \underline{x}_t)$ would then indicate the state estimation capability of this combined estimate.

However, the state estimation performance of the 3-dimensional attitude filters is poor because of simplified propagation models within the filters. Therefore, such an "overall-optimal" estimate is not warranted. If a more sophisticated propagation model were incorporated, this would be a viable concept. Such a sophisticated model would be of higher dimension than three, so equations (74) and (76) would be computationally burdensome. One could then utilize the approximation of equation (75) or use the relationships

$$\hat{\underline{x}}_{OVERALL} = \hat{\underline{x}}_{INS} + \underline{K}_o \left[z_{AHRS} - \underline{H} \, \hat{\underline{x}}_{INS} \right] \qquad (77)$$

$$\underline{K}_o = \underline{P}_{INS} \, \underline{H}^T \, [\underline{H} \, \underline{P}_{INS} \, \underline{H}^T + \underline{R}_{AHRS}]^{-1} \qquad (78)$$

$$\underline{P}_{OVERALL} = \underline{P}_{INS} - \underline{K}_o \, \underline{H} \, \underline{P}_{INS} \qquad (79)$$

where the inverse in equation (78) is a (3 x 3) matrix inversion. That these equations duplicate equations (74) and (76) can be proven by methods presented in Reference [ ].

The variable $\underline{z}_{AVG}$ mentioned in the fourth set of outputs is the weighted average of the measurements taken from the INS and AHRS:

$$\underline{z}_{AVG} = \left[\underline{R}_{INS}^{-1} + \underline{R}_{AHRS}^{-1} \right]^{-1} \left[\underline{R}_{INS}^{-1} \, \underline{z}_{INS} + \underline{R}_{AHRS}^{-1} \, \underline{z}_{AHRS} \right] \qquad (80)$$

If $\underline{R}_{INS}$ and $\underline{R}_{AHRS}$ are diagonal matrices (as they often are), then a relationship similar to (75) would be exact, rather than an approximation:

$$z_{AVG_i} = \frac{1}{R_{INS_{ii}} + R_{AHRS_{ii}}} \left[R_{AHRS_{ii}} \, z_{INS_i} + R_{INS_{ii}} \, z_{AHRS_i} \right] \qquad (81)$$

Such an evaluation would be a best estimate of the Euler angles based on both INS and AHRS data, useful in the event that a failure were to affect the AFCS rate gyros.

The fifth set of outputs are printouts and plots of the likelihood function values over a given computer run. Especially useful are the plots of the individual likelihood functions, since the stead,-state and transient characteristics of those functions will be of utmost importance in the declaration of failed sensors. The distinguishing aspects between the likelihood functions resulting from normal operation and those generated when a failure has occurred can be more readily discerned from time plots than from data printout. In fact, it was the use of these plots that enabled this investigation to improve the failure detection capabilities of the functional redundancy method so markedly. Understanding the dynamic characteristics of the likelihood functions suggested the incorporation of "time-to-failure-declaration" parameters in conjunction with thresholds for each likelihood function, both of which are also printed outputs of the software package.

Besides outputting the individual likelihood functions (each of which is an N-step sum), the software also prints out the value of the individual terms that compromise the likelihood functions, i.e., terms of the form $\{-1/2[e^2(i)/\sigma^2(i)]\}$. Also printed are the individual values of $e^2(i)$, the squared value of the observed residual, and of $\sigma^2(i)$, the variance of the residual sequence as propagated by the filter itself. If and when large magnitude likelihood functions or other off-nominal characteristics occur, then these outputs aid the analysis of their generation.

The final selection of appropriate values for the thresholds and time-to-failure-detection parameters for each likelihood function is expedited by observing the minimum and maximum likelihood function magnitude in the last I iterations, where I is an adjustable integer. Here I is actually a proposed value for the number of algorithm iterations before a failure is declared. By looking at both no-failure and sensor failure test cases, the threshold and I values can be chosen so that (ideally) no normal operation will cause the likelihood function to exceed the threshold for I consecutive iterations, while (ideally) all appropriate failure cases will cause this threshold to be surpassed for at least I iterations consecutively.

The periodic display of all pertinent simulation or real environmental data, in addition to model reference and likelihood function performance information, is performed for convenience of the user. Whether or not this display is made, and its frequency of occurrence, can be controlled.

2.4 FILTER TUNING

As described previously, the Kalman filters are tuned by adjusting $\underline{Q}_{ADJ}$ and $\underline{R}_{ADJ}$ until the sequence of $(\hat{\underline{x}} - \underline{x}_t)$ values in each filter correlates well with the standard deviations propagated in the filter $\underline{P}$ matrices during a set of Monte Carlo runs. This adjustment is an iterative trial and error process, but some guidance can be suggested for the procedure.

First of all, the sequence of differences between the simulated output of any sensor and the simulated "true" value of the variable should be analyzed to verify that the established values of $\underline{Q}_{SENSOR}$ and $\underline{R}_{SENSOR}$ correlate reasonably. If real data is used instead of simulated data, some calibration period or other similar condition will provide sensor output during which time the true value of the measured parameter is known, and a similar procedure can be followed.

If the $\underline{P}$ matrix underestimates the error statistics, especially during periods of significant maneuvering, i.e., in a transient manner, the values of $\underline{Q}_{ADJ}$ rather than $\underline{R}_{ADJ}$ should be increased. Thus, if the error in the roll estimate exceeds the level predicted by $\underline{P}$ for a time interval after a roll maneuver, then the corresponding element in $\underline{Q}_{ADJ}$ would be increased to show a decreased confidence in the ability of the filter dynamic model to represent the physical situation adequately. This adjustment should be coordinated with the effects on the appropriate likelihood function plots. This sensitivity to trajectory dynamics is treated more fully in paragraph 2.6 of this section.

In the case of the two attitude filters, each driven by the same rate gyro information, if the characteristics with which $\underline{P}$ misrepresents the true error covariance are very similar in the two filters, then adjustment of $\underline{Q}_{ADJ}$ is appropriate. However, if only one $\underline{P}$ misrepresents the error covariance significantly, then its associated $\underline{R}_{ADJ}$ should be compensated.

2.5 THRESHOLDS AND TIME-TO-FAILURE-DECLARATION PARAMETERS

Standard procedure for setting thresholds for likelihood function hypothesis testing would be to conduct a number of trials with no failures and determine the largest magnitude attained by each of the likelihood functions. Then a series of failure runs would be made, and the minimum magnitudes of likelihood functions that are expected to demonstrate a sensitivity to a certain failure are recorded. If a region of uncertainty is thereby established, i.e., if there exist some likelihood function magnitudes below the largest magnitude achieved with no failures while above the smallest magnitudes attained with pertinent failures, then some compromise is necessary. It might be appropriate to set the threshold so as to preclude either false alarms or missed alarms (not both), or to choose a threshold level between these two extremes and accept some percentage of both missed alarms and false alarms.

Considerable effort was expended in an attempt to make threshold setting more methodical, rather than simply looking at highest likelihood values attained under normal conditions and the lowest attained under failed conditions. What resulted was a means of predicting the probability of detection and probability of missed alarm when a sensor failed, as a function of the threshold setting. Similarly, the probabilities of no alarm or a false alarm when a failure did not actually occur could also be calculated as a function of threshold level.

Such a description of the probability of the detection logic signalling a failure is developed in the following manner. A failure is declared if the likelihood function becomes more negative than some threshold level; i.e., if

$$L_N(i) < -T \tag{82}$$

where T is the magnitude of the threshold and $L_N(i)$ is approximated as (see Section II.1.4):

$$L_N(i) \cong \sum_{j=i-m+1}^{i} \left\{ -\frac{1}{2}\left[e^2(j)/\sigma^2(j)\right] \right\} \tag{83}$$

When a Kalman filter model reference is used, e(j) is one of the filter residuals and $\sigma^2(j)$ is the filter's estimate of the variance of that residual error, as given in equations (43) and (44). Thus a failure is declared if:

$$\sum_{j=i-m+1}^{i} \left[e^2(j)/\sigma^2(j)\right] > 2T \tag{84}$$

First consider the simplest case of N = 1. Then equation (84) relates that a failure is declared if

$$|e(i)| > \sqrt{2T}\quad \sigma(i) \tag{85}$$

Now assume that the actual residuals, under either no-failure or failed conditions, can be described (or at least approximated) statistically by a Gaussian density with mean b(i) and variance $\sigma_t^2(i)$. Then the probability of declaring a failure is the shaded area in Figure 4. If this plot is normalized by using σ_t as a scaling factor, as in Figure 4, then the probability of declaring a failure can be computed from unit normal density tables for selected numerical values for (b/σ_t) and $(\sqrt{2T}\ \sigma/\sigma_t)$.

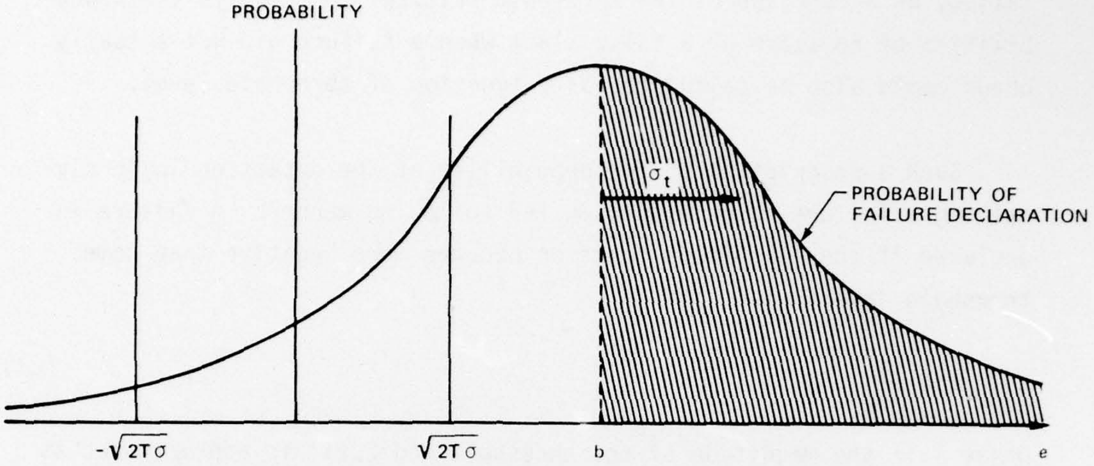


Figure 4a. Probability of Failure Declaration

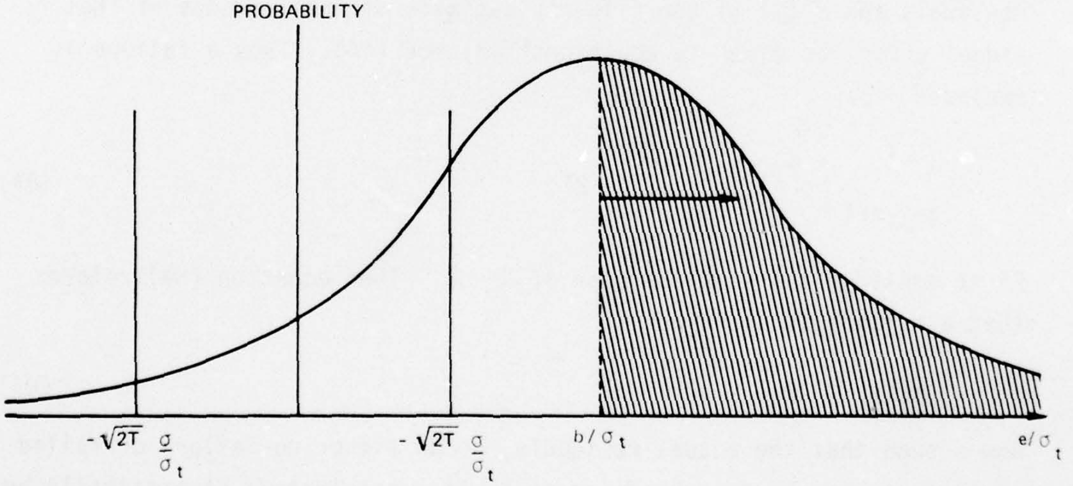
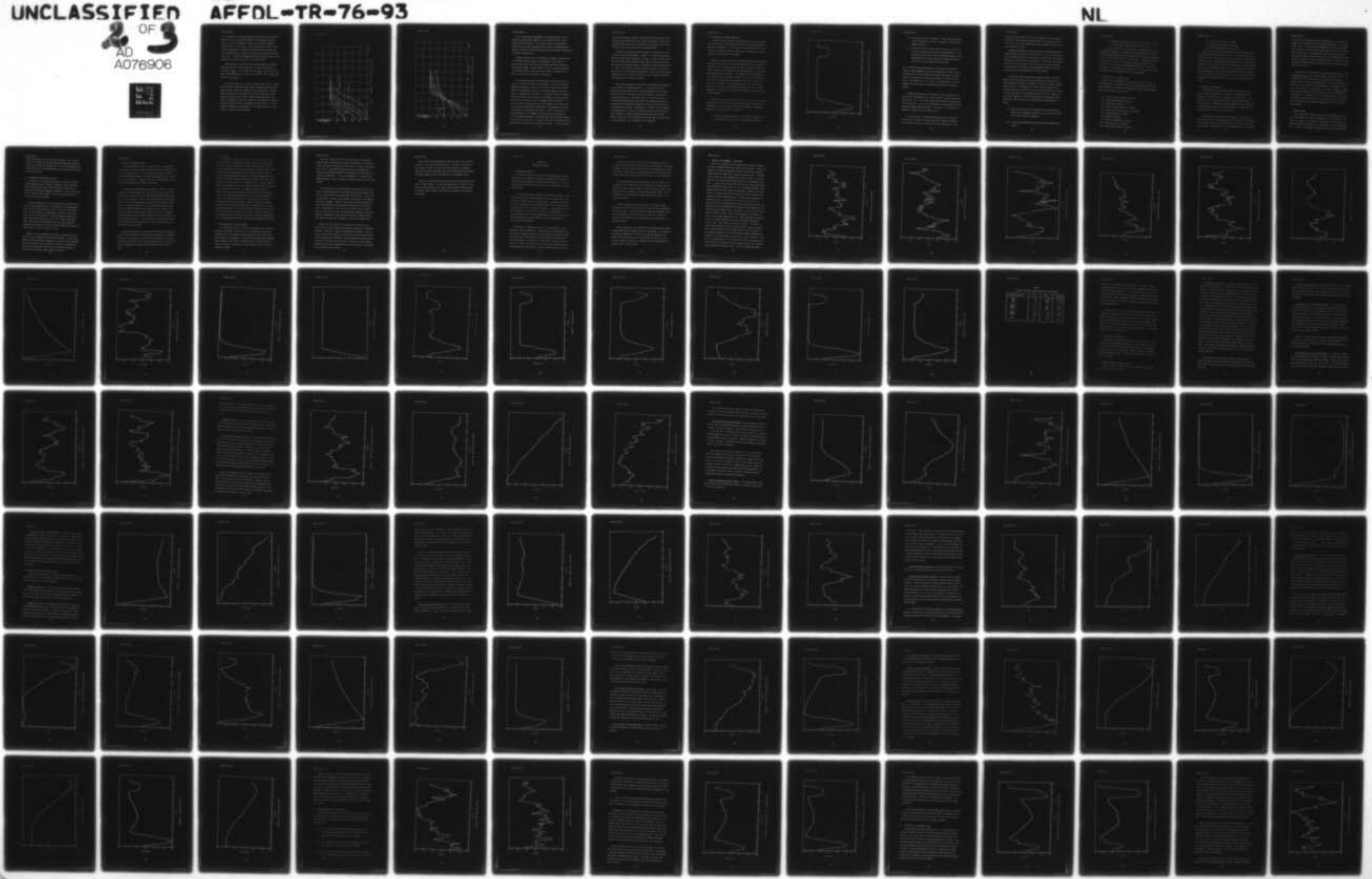


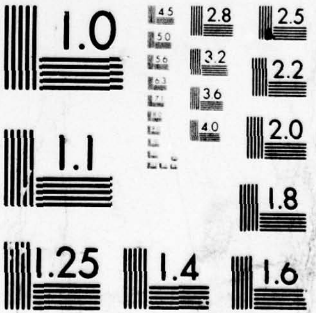
Figure 4b. Normalized Density

AD-A076 906   AIR FORCE FLIGHT DYNAMICS LAB WRIGHT-PATTERSON AFB OH   F/G 1/3
FAILURE DETECTION THROUGH FUNCTIONAL REDUNDANCY.(U)
SEP 76 P S MAYBECK

UNCLASSIFIED   AFFDL-TR-76-93   NL

2 OF 3
AD
A076906




1.0
4.5
5.0
5.6
6.3
2.8
3.2
3.6
4.0
2.5
2.2
2.0
1.1
1.8
1.25
1.4
1.6

It is then possible to plot the probability of declaring a failure as a function of b/σ_t, for different values of the normalized threshold $\sqrt{2T}\ \sigma/\sigma_t$. This is presented in Figure 5, for normalized threshold values of 0.5, 1, 1.5, 2, and 3. Also plotted on this figure is the locus of probabilities of declaring a failure when the actual mean of the residual density, b, assumes the critical value for failure declaration, $\sqrt{2T}\ \sigma$. (Note that an actual failure would probably cause b to be greater than this value or causes σ_t to be very large.)

Referring back to Figure 4a, another useful means of presenting this data would be to plot probabilities of failure declaration as a function of $(\frac{b}{\sqrt{2T}\ \sigma})$. This is possible, since $\frac{b}{\sqrt{2T}\ \sigma} = (b/\sigma_t)/(\sqrt{2T}\ \sigma/\sigma_t)$ and is depicted in Figure 6. Here, the locus of $b = \sqrt{2T}\ \sigma$ is along the vertical line at $\frac{b}{\sqrt{2T}\ \sigma} = 1$.

For N greater than one, a similar procedure would be used. Assume that the filter estimate of variance does not change significantly in N sample periods, that the N residuals are each described by a Gaussian density with mean b and variance σ_t^2 as before, and that the N values formed by subtracting b from each residual are uncorrelated. Then a parallel development is possible, except that the probability density of a Gaussian random variable is replaced with the density of a chi variable with N degrees of freedom [Ref. 4]. For each value of N, a plot similar to Figure 5 can be developed.

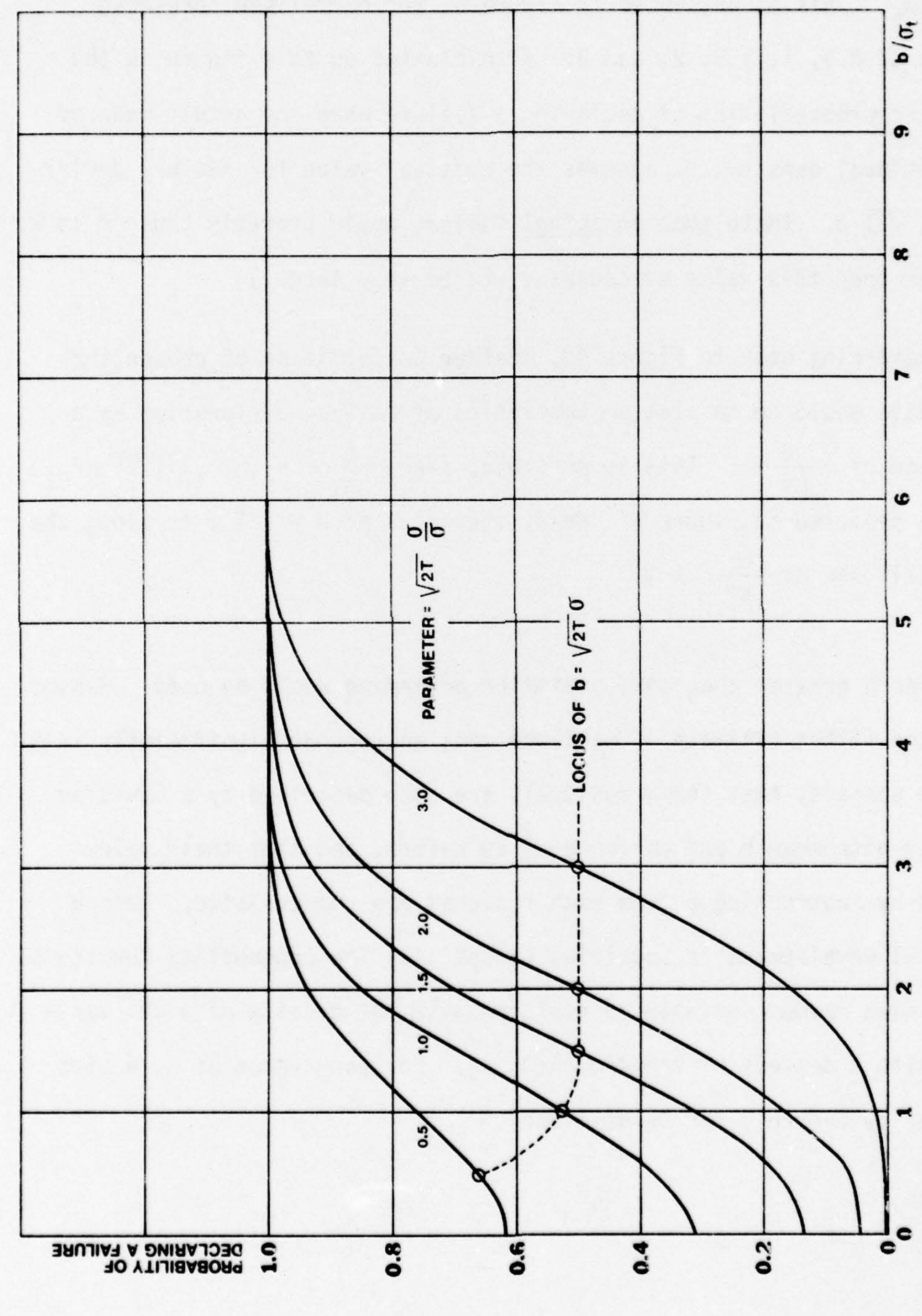


Figure 5. Probability of Failure Declaration vs b/σ_t

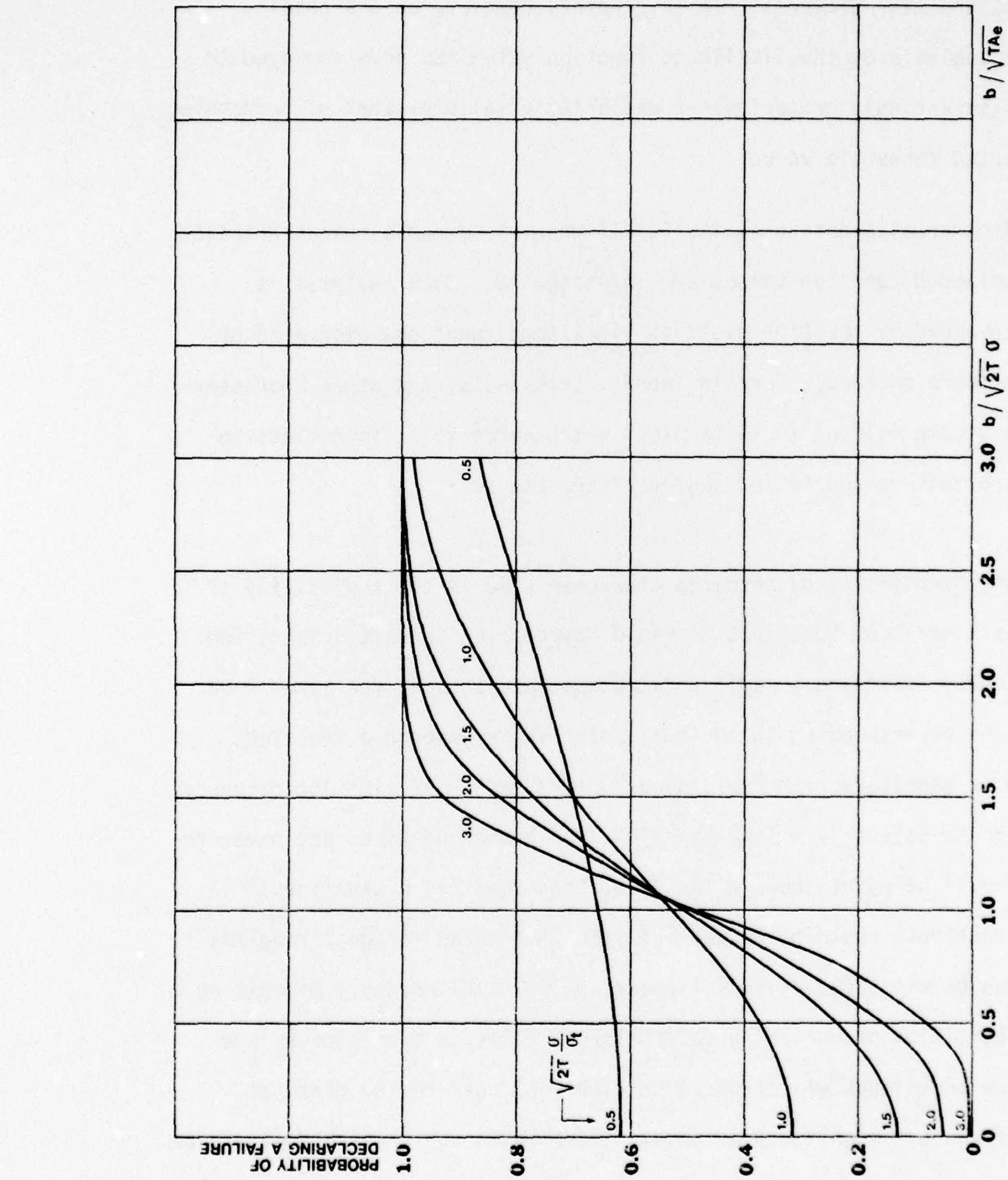


Figure 6. Probability of Failure Detection vs $b/(TAe)^{V2}$

Thus, a more complete knowledge of system performance can be attained by predicting the probabilities of detected failures, false alarms, and missed alarms. To this point, however, an essentially static analysis of the likelihood function value has been employed to establish a single criterion for declaring a failure--that of surpassing a selected threshold value.

Another alternative avails itself when the dynamic characteristics of likelihood function values are investigated. Such analysis is greatly aided by the time plots of likelihood functions generated by the software package. Certain trends, transients, and other characteristics become evident in these plots which serve to differentiate between no-failure and failed sensor circumstances.

One significant discernible characteristic is the sensitivity of certain likelihood functions to rapid changes in aircraft orientation. Immediately following a rapid roll to initiate a turn, the likelihood functions corresponding to the roll Euler angle undergo a transient growth in magnitude with rapid decay. (The length of time for recovery from the transient is a function of N, the number of times any given residual will be maintained in the likelihood function evaluation.) If such transients could be masked out, the thresholds for declaring failures can be set substantially tighter, while simultaneously minimizing the probability of declaring false alarms. This is true because the standard techniques of threshold setting would record the peaks of these transient magnitudes as levels above which the threshold should be set to preclude false alarms. Actually, the typical likelihood function magnitudes are substantially lower than these peaks. If the transients could be recognized and removed, the maximum likelihood function magnitude under normal conditions, other than rapid transients, and thus an appropriate threshold level for preventing false alarms, would be significantly lower than achieved by the standard approach.

One means of masking out these transients is through use of time-to-failure declaration parameters. Suppose it is known that transients occurring during normal operation will surpass a certain threshold level but will rapidly return below that level, while any pertinent failure will cause the likelihood function to surpass the level and remain above it. Then it is possible to establish a failure detection criterion of the form, "If the likelihood function passes a given threshold level and remains above it for a specified period of time (or number of algorithm iterations), then a failure is declared."

Setting these thresholds and time-to-failure-declaration parameters must be done in a coordinated fashion. If a long time-to-failure-declaration parameter were chosen, a tight threshold could be chosen with few false or missed alarms, but at the expense of a delay in declaring real failures. On the other hand, if a very short time-to-failure-declaration parameter were chosen, the ambiguity between likelihood function transients and behavior due to real failures would not be substantially decreased. Consequently, a trade-off must be conducted and the best pair of values (of a threshold and time-to-failure-declaration parameter) for each likelihood achieved after some iterative search. Note that, for a likelihood function that does not exhibit such transient behavior, only a threshold value is required.

2.6 SENSITIVITY TO AIRCRAFT MANEUVERS

The sensitivity of certain likelihood functions to aircraft maneuvers was mentioned as the primary motivation for the time-to-failure declaration parameters. In this section, an elaboration of this sensitivity will be made, indicating causes, means of alleviation, and effects upon failure detection philosophy.

Figure 7 is a typical plot of the likelihood function corresponding to the roll Euler angle in the AFCS-INS Kalman filter in a no-failure simulation run. The two transient dips occur at the times when the aircraft first rolls to initiate a turn and then again when it rolls to resume straight-and-level flight. It is noted that the likelihood function employed was a 10-step sum of terms (i.e, N = 10), and that both the rate of recovery from such transients and the ratio of transient peak value to "normal" likelihood function value are a function of N. During the turn itself, the likelihood function returns to a "normal" magnitude; it is only the rolling maneuvers themselves that generate the transient behavior.

This behavior can be attributed to the inability of the first order integration of simplified nonlinear equations to model adequately the true dynamics of a rapid change in aircraft orientation. Therefore, some means of reducing the effect would be:

(1) improving the dynamics model, at the expense of higher dimensional state vectors in the filters and computer loading;

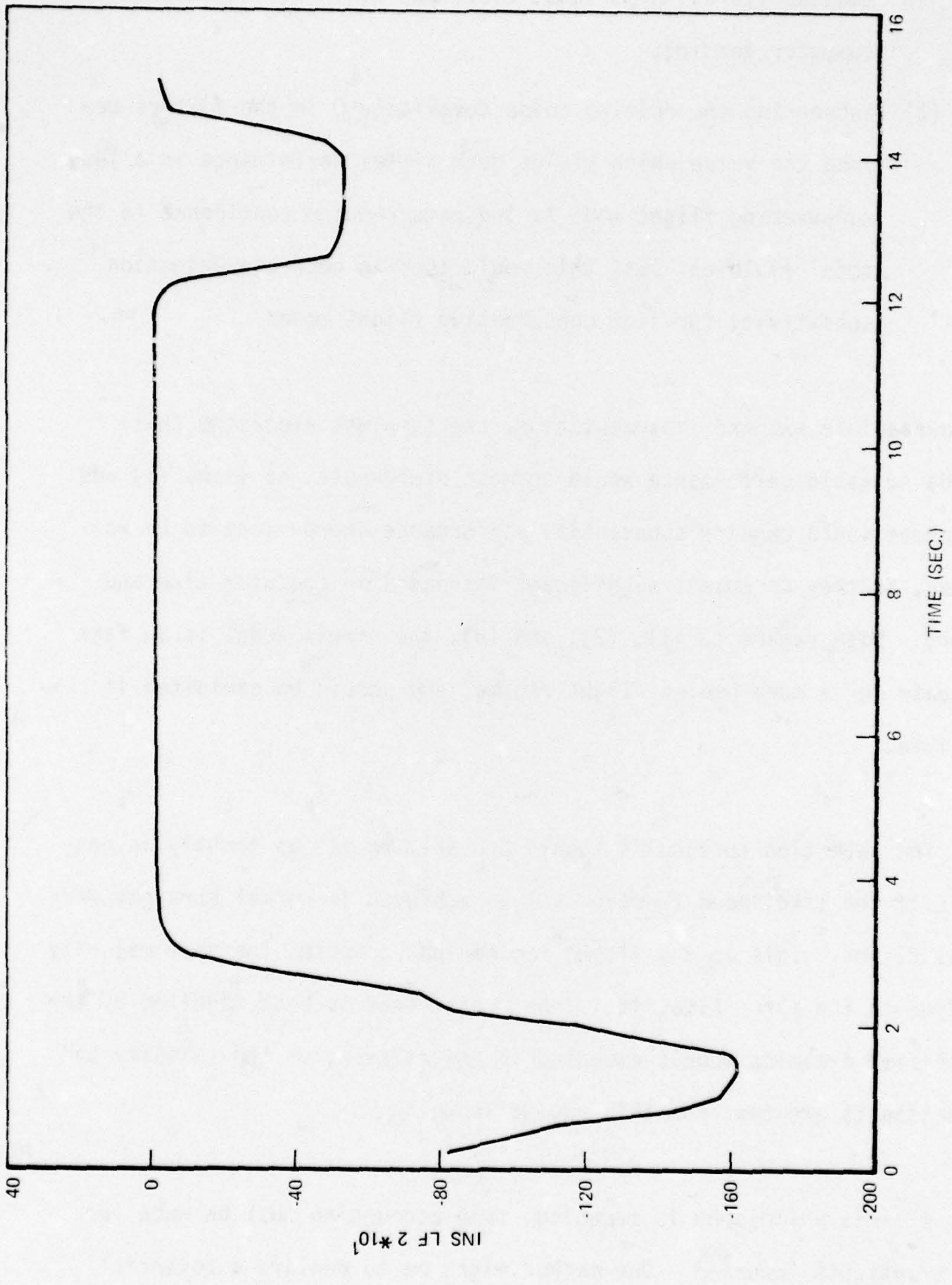


Figure 7. Sensitivity of Roll Likelihood Function to Rapid Roll Rates

(2) improving the means of integration: higher order techniques, smaller iteration periods, etc., but again at the expense of computer loading;

(3) increasing the driving noise covariance $\underline{Q}$ in the filters beyond the value which yields good filter performance in a less maneuvering flight mode to indicate reduced confidence in the model employed; but, this would tend to decrease detection sensitivity for such conservative flight modes.

For a feasible onboard implementation, the simplest algorithm that yields suitable performance would be most preferable, so items (1) and (2) above would require substantial performance improvement to be warranted, as they do entail significant increases in computer time and memory. With regard to (1), (2), and (3), the simple model is in fact adequate for a more benign flight regime, and should be exploited if possible.

The detection thresholds should probably be set as tightly as possible to the likelihood function values achieved in normal straight and level flight. This is the flight regime that composes the vast majority of time in the air. Also, it is the regime that is best modelled by the simplified dynamics models embodied in the filters, so the validity of detection is greatest for this regime as well.

If this philosophy is accepted, some accounting must be made for the transients incurred. One method might be to declare a potential failure, remove the sensor from the filter inputs (especially if the filter can operate without it), but allow immediate recertification of the sensor. Besides causing nuisance alarms, this technique would result in the loss of some valuable data before recertifiction if a failure did not really occur.

Another method would be to use the time-to-failure-declaration parameter concept as described in the last section. This is a rather simple and effective solution to the problem, but does suffer from causing some delay in declaration of actual failures. Such a delay would result not only in use of bad data by the aircraft control system, but might also cause filter performance to diverge beyond the point of recovery once the bad data were removed from its input channels.

If the time-to-failure-declaration parameter concept is not adequate, making the failure detection logic adaptable to the amount and type of maneuvering might be considered. By monitoring control surfaces such as ailerons, or commands sent to these control surfaces by the pilot and autopilot, the detection logic could know when high roll rates or other transient-inducing phenomena were going to occur. Under "normal" flight conditions, the logic would employ the appropriately tight thresholds. When informed of such transient-inducing phenomena, it could

(1) simply nullify any failure declarations due to threshold passage until the phenomena terminated (as, until high roll rates are no longer sensed or commanded),

(2) invoke higher magnitude thresholds until the phenomena terminated, or

(3) invoke the additional time-to-failure declaration criterion besides threshold passage during this time period.

It should be mentioned that one reason for conducting flight tests to generate real data is to corroborate the characteristics of Figure 7. There was some question about the realism of roll angle values generated by the simulation program during a roll maneuver. Although the same basic trend of this figure is anticipated, the exact character of the transients may well be less pronounced using real data. Nevertheless, this effort has assumed that the simulation is valid in order to conduct performance analyses (subject to revision if necessary).

2.7 SENSITIVITY TO INSTRUMENT BIASES

Sensitivity of performance to in-tolerance instrument biases is an important concern. Therefore, twenty-four separate biases are individually adjustable in the design tool-performance analysis program package. These are the separate biases on the:

(1) three Euler angle outputs of the INS

(2) three accelerometers of the INS

(3) three gyros of the INS

(4) three Euler angle outputs of the AHRS

(5) two axes of the vertical gyro of the AHRS

(6) directional gyro of the AHRS

(7) compass of the AHRS

(8) normal accelerometer of the AFCS

(9) three rate gyros of the AFCS

(10) static pressure signal of the ADS

(11) pitot pressure signal of the ADS

(12) angle of attack output of the ADS

(13) total temperature output of the ADS.

Both statistical and noise case descriptions of these biases were developed for the equipment used in the F-4. With these values, realistic effects due to in-tolerance sensor biases could be analyzed by the software package. Computer runs were conducted with all biases zeroed except for one (to study sensitivity of performance to individual biases), and all set to representative values (to investigate combined effects). By determining which individual bias variations cause the most degradation in performance, one can specify which sensors must have the tightest bias drift characteristics for utilization in the integrated data system.

2.8 PREFLIGHT INITIALIZATION

A data reduction program capable of efficient calculation of the mean, variance, and whiteness of a sequence of sampled data signal values was described earlier in paragraph 2.2 of this section. One application of this program would be for preflight initialization. As envisioned here, a standard test computer program could be implemented in ground support equipment (or possibly onboard) to produce good initialization before each flight of the vehicle.

First of all, by running the sensor systems in a preflight test when the true values of measured parameters are available, the computed mean of signal values can be used to estimate the biases in the individual sensors. From analysis of sensor performance data it can be seen that typically the turnon-to-turnon nonrepeatability is considerably greater than instrument bias drift generated during the mission flight period. Thus, if the biases were estimated each time the instruments were turned on, then these bias estimates would be valid for the endurance of the mission. Such compensation would improve sensor system performances, decrease concern about sensitivity of detection to biases, increase the adequacy of the simple dynamic models in the filters, and substantially reduce the need of adding on-line bias estimation capability to these filters.

Furthermore, the estimation of variance could be exploited as well. The $\underline{R}_{SENSOR}$, $\underline{Q}_{SENSOR}$, and $\underline{P}_o$ values embodied in the Kalman filters are established by statistical testing of representative instruments. Preflight analysis of sensors could determine if the particular sensors onboard the aircraft perform in the same manner as the "population statistics" would indicate. In other words, valid values of $\underline{R}_{SENSOR}$, $\underline{Q}_{SENSOR}$, and $\underline{P}_o$ could be obtained for each individual aircraft's complement of instrumentation. In addition to this fine tuning to particular sensors, such analysis performed routinely on the same aircraft over a period of time could indicate aging and other performance trends of the instruments onboard.

3. MODES OF USAGE

Once the functional redundancy logic has been developed with the aid of the design tool, onboard implementation can be considered. This logic is not meant to be a detection system unto itself, but part of an integrated failure detection system, as described in the next paragraph (3.1). Section 3.2 subsequently considers the various appropriate means of declaring failures with the functional redundancy logic. Then Section 3.3 investigates how the data system might adapt and reconfigure itself, once a failure has been declared. As will become apparent, the choices made here can yield implementations that range from very simple to very sophisticated.

3.1 INTEGRATED FAILURE DETECTION SYSTEM

As mentioned in the beginning paragraphs of Section I, the functional redundancy concept is meant to complement, rather than totally replace, other means of sensor failure detection. By being used in conjunction with hardware redundancy, deterministic tests, built-in-test (BIT), and ground support methods, an efficient, integrated, failure detection system can be achieved.

Functional redundancy is not the most appropriate technique for the entire failure detection system. For instance, BIT and reasonableness tests can readily detect many hard failures with very little computation. On-line estimation and compensation of biases and scale factor errors are more easily achieved by hardware redundancy, though functional redundancy can provide an "extra voter" in the original detection. Also, because of the response time of the logic, functional redundancy would probably not be the sole means of detecting failures of sensors that are critical to safety of flight.

However, functional redundancy does provide a substantial contribution to such an integrated failure detection system. It significantly reduces the required hardware redundancy for attaining "two-fail-operate" capabilities or other similar degrees of reliability. By correlating data from different types of data sensors, it removes the need for a proliferation of identical sensors onboard an aircraft.

3.2 MEANS OF DECLARING FAILURES

In the simplest form, a failure is declared if a likelihood function surpasses a given threshold (which may or may not be adaptively set to aircraft maneuvering). For some likelihood functions, time-to-failure-declaration parameters are also incorporated into the criterion for failure declaration. However, there are certain additional aspects of failure declaration that should be considered.

If a time-to-failure-declaration parameter is used with a certain threshold for failure detection, it may be desirable to be able to detect obvious failures without the inherent delay caused by that parameter. For that reason, a second, larger threshold might be established such that if the likelihood function surpasses both thresholds, then a failure is declared immediately. Such a multiple threshold could be used to discriminate between hard failures (in which no useful data would be expected) and soft failures (which could result in degraded sensor performance, but some useful information still is expected to be available from the sensor). This discrimination capability might warrant different data system adaptations to hard and soft failures, as discussed in the next section.

Multiple thresholds might also be used to advantage in another way. A smaller threshold might be established and time beyond that threshold recorded as an indication of a sensor starting to go out of tolerance, or degrade in some other manner, without actually failing. Such a test could be conducted with ground equipment rather than on-line at all times.

By referring to Table IV in Section II.2.6, it can be seen that failure isolation is dependent upon which of the ten likelihood functions are beyond threshold value. Certain failures affect a single likelihood function while other failures affect a number of them. Thus, if a failure that affected two or more likelihood functions were to occur, and the thresholds were passed at slightly different times, it is possible to mistakenly declare a failure that affects the single likelihood function that surpasses its threshold first. For that reason, it might be advantageous to signal an alarm that some failure has occurred, giving the possible failures. The probability that the failure is actually one that affects multiple likelihood functions can be determined by monitoring the magnitudes of the other likelihood functions. If they are of normal magnitude, the failure that affects only one likelihood function can be declared. However, if they too are above normal magnitude, failure isolation might be delayed for one or a few algorithm iterations, to be more sure of what failure did in fact occur. In the interim period, the most likely failure might be displayed, or all possible failures shown with an indication of the most probable one, or no indication made other than a signal that some failure has occurred.

3.3 LOGIC ADAPTATION TO FAILED SENSORS

If a sensor has failed, its inputs to the detection logic would be removed, as described in Section II.2.8. The theoretical question of observability of the resulting filter models when various sensors have been removed was investigated. It was also mentioned that such degraded model references could provide estimated values of INS or AHRS Euler angles, AFCS normal acceleration, or ADS altitude, vertical velocity or indicated airspeed.

Removal of sensor signals from the rate gyros to the attitude Kalman filters, or from the INS vertical accelerometer to the vertical filter, precludes the operation of those filters. Unless a hard failure were to occur, a degraded performance mode could be attempted by increasing the appropriate elements of the $\underline{Q}$ matrices in the filters and accepting the poor data. This form of adaptation is included in the software package, as well as removal of failed sensors from filter input channels.

If soft failures can be distinguished from hard failures, as mentioned in the previous section with regard to multiple thresholds, then "failed" sensors need not be removed from the data system. A hard failure would result in sensor signal removal, with zeroing of the appropriate $\underline{H}$ matrix elements. However, a soft failure could be handled by increasing the magnitude of the appropriate element in the $\underline{R}$ covariance matrix. Additionally, if the failure can be identified as a stable shift in sensor output, rather than more random fluctuations, then some attempt at compensation of the soft failure might be conducted (using other means than functional redundancy to achieve the compensation).

However the logic adapts to the failed sensor, there will be a certain period of time required for the model references and likelihood functions to recover to "normal" (but degraded) performance. For this reason, when a failure does occur, the ability to declare other failures should be temporarily inhibited so as not to generate many false alarms. An indication could be sent to the pilot to inform him of the extent of degraded sensor systems, degraded failure detection ability, and reduced mission capabilities that result.

Once a sensor has been declared as having failed, it may be useful to monitor it for possible recertification. Especially in the case of soft, stable failures, as biases whose values could be estimated by other means and then compensated, such recertification may be warranted. This would, however, add to the complexity of the detection logic.

There are a variety of means of declaring failures and adapting the logic to those failures. For any given application, a trade-off of complexity versus performance gain would be required before deciding upon the eventual mode of usage of the functional redundancy failure detection concept.

SECTION IV
EXPERIMENTAL RESULTS

1. EXPERIMENTS CONDUCTED

This section presents the results of an extensive series of performance analysis simulations conducted to demonstrate the capabilities of the functional redundancy failure detection algorithm. It will be seen that the feasibility and efficiency of the algorithm has, in fact, been verified.

By first generating all required initial conditions through a long trajectory simulation, three evaluation trajectory segments were established: a straight-and-level flight segment, a trajectory composed of a roll into a coordinated turn followed by another roll to resume straight-and-level flight, and a pitchover and descent. As mentioned previously, this simulation was "flown" by an F-4 with a full complement of particular sensors normally carried onboard such an aircraft, except that the INS characteristics were modified to be more representative of state-of-the-art technology.

First of all, a Monte Carlo set of runs was performed in order to establish a basis of comparison. Two different sets of Monte Carlo runs were actually conducted, one set in which instrument biases were allowed to assume various representative values and another set in which all biases were zeroed. The latter set is used later in the bias sensitivity tests. During these first test runs, the filters were tuned to the straight-and-level flight profile. Section 2 delineates these results.

Monte Carlo runs of the same trajectory segments were then conducted, but with individual sensor failures being simulated. Both types of Monte Carlo runs, with zeroed biases and representative bias values, were used. The resulting likelihood function plots then allowed evaluation of the various means of declaring failures, as seen in Section 3.

Since one problem was the sensitivity to rapid roll rates, an investigation into altering the filter driving noise covariances ($\underline{Q}$) to tune the filters to an environment of higher roll rates was made. This detunes the filter somewhat in the straight-and-level regime. The effectiveness of the $\underline{Q}$ variance to reduce maneuver sensitivity is presented in Section 4.

Section 5 then presents the sensitivity to instrument biases. Monte Carlo runs with all biases set to zero but one were performed for each bias in turn. To make the effects of the biases pronounced, each one being tested was set at the worst case level (or 2σ value if a statistical description of bias characteristics was available for a particular sensor).

Section 6 describes the verification of the simulated data results by data tapes acquired through flight test recordings. Due to some extenuating circumstances, the flight tests have been delayed for 21 months, and so the actual data is not available at the time of this writing. However, the tests are scheduled and this substantiation is expected within the near future.

2. BASELINE OF PERFORMANCE - NO FAILURES

Monte Carlo simulations were conducted over the three evaluation trajectory segments without any failures being simulated. Figure 8 presents plots of the ten likelihood functions for the three trajectory segments under the conditions of zeroed instrument biases. The labels in this figure are used throughout this section: INS 1, 2, and 3 are the pitch, roll, and yaw likelihood functions for the INS-AFCS filter; AHRS 1, 2, and 3 are similarly pitch, roll, and yaw likelihood functions for the AHRS-AFCS filter; VERT 1, 2, and 3 are lagging altitude, independent vertical velocity (used to check angle of attack), and vertical velocity likelihood functions of the vertical filter; and VIAS is the indicated airspeed likelihood function. These labels correspond to e_1 through e_9, and e_{11}, respectively, of Table IV in Section II.2.6. Variation of these instrument biases did not substantially increase the magnitude of any likelihood function except the one corresponding to indicated airspeed (labelled VIAS in the plots). In fact, some effort was made to include in Figure 8 plots of large magnitude likelihood functions from the various Monte Carlo runs. Maximum magnitudes attained by the ten likelihood functions on the three trajectories (denoted as "level," "turn," and "descent") are given in Table V. This table includes nonzero bias runs as well as zero bias runs. With zero biases, after a rapid transient from an initial value of -195 as in Figure 8, the maximums attained by the VIAS (indicated airspeed) likelihood function were -10 in level flight, -20 in the turn, and -8 in descent. Note that this original value of -195 was due to the initial conditions being established with a trajectory in which instrument biases were allowed to assume representative nonzero values. These are substantially lower

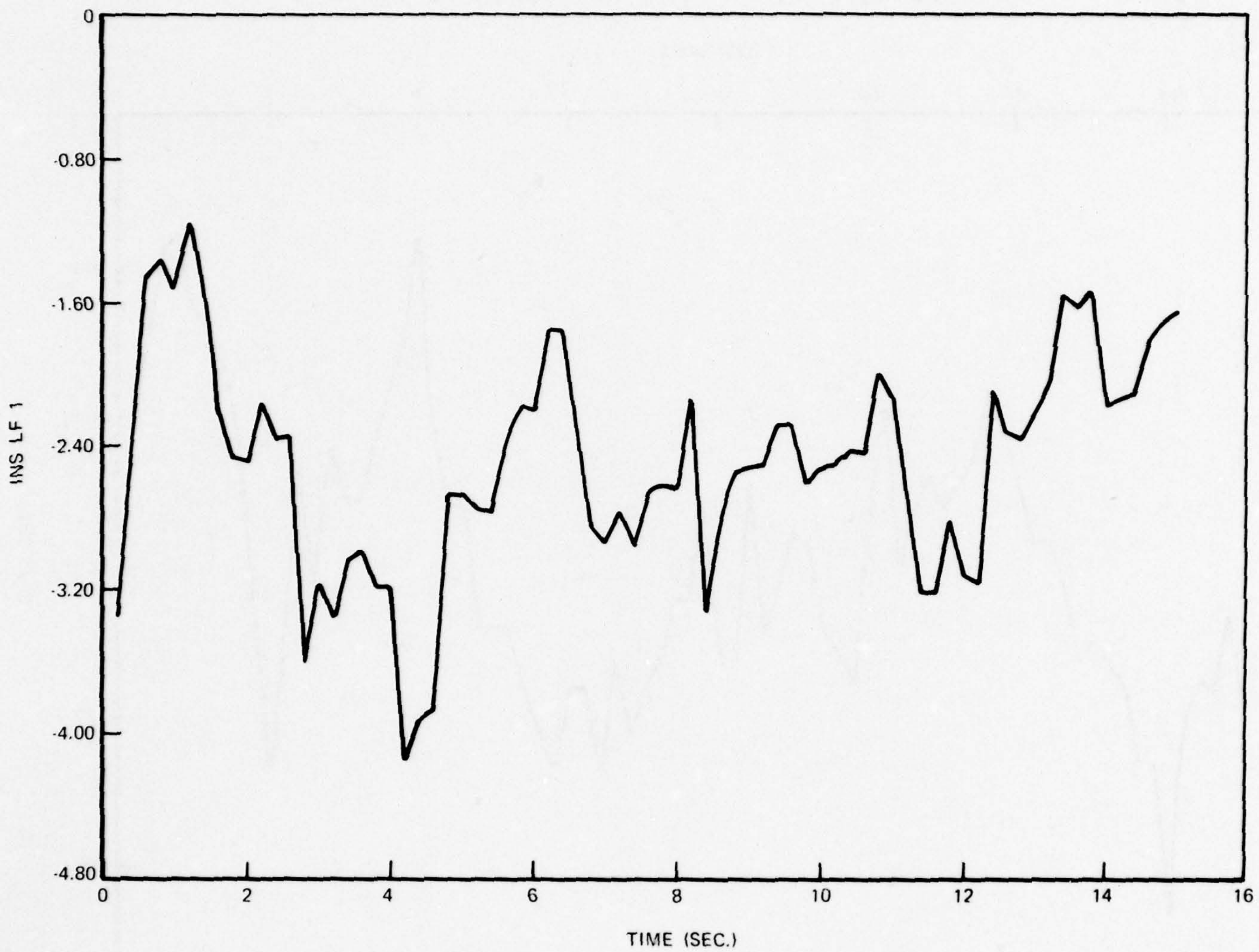


Figure 8a. No Failures; Level Flight

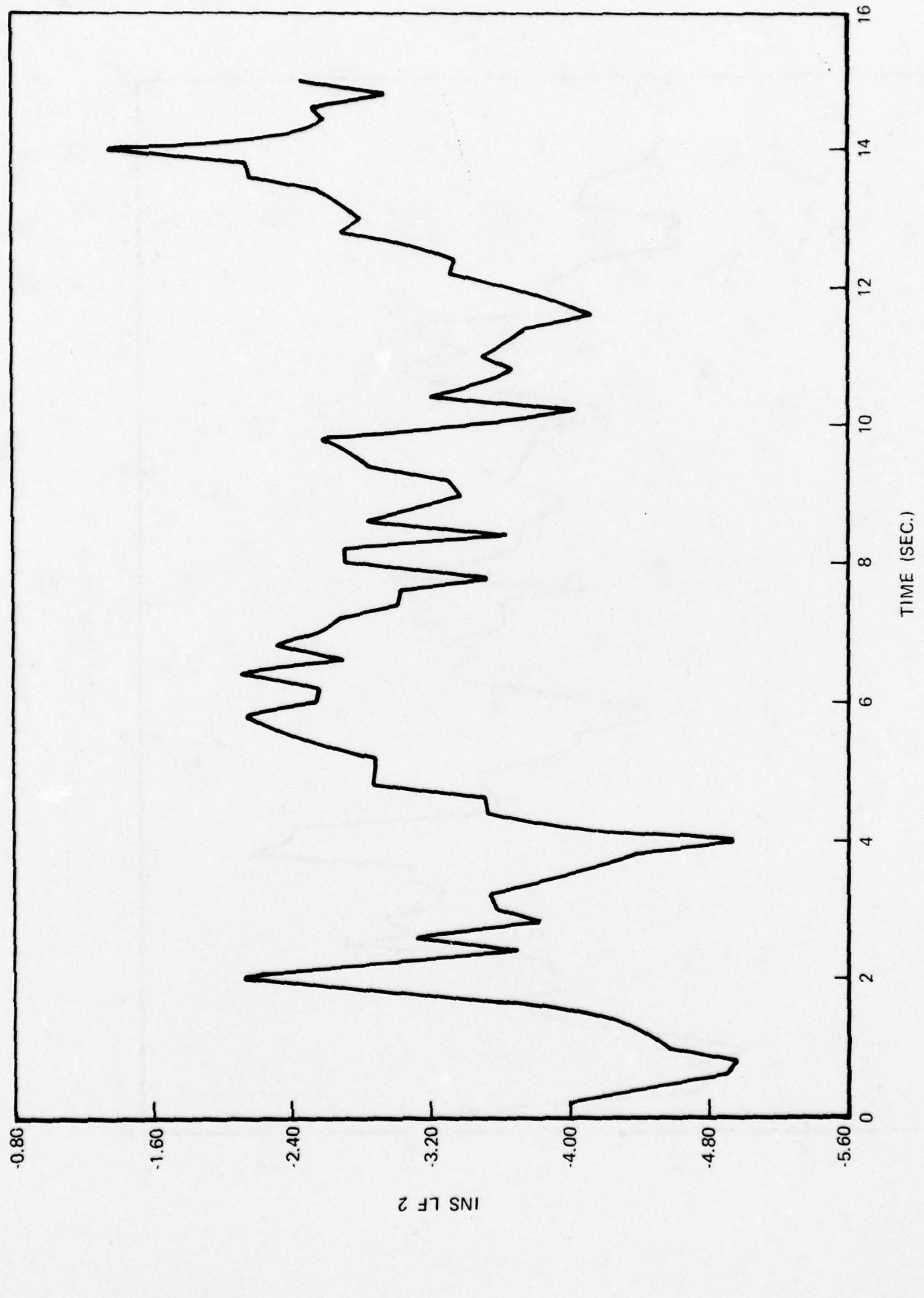


Figure 8b. No Failures; Level Flight

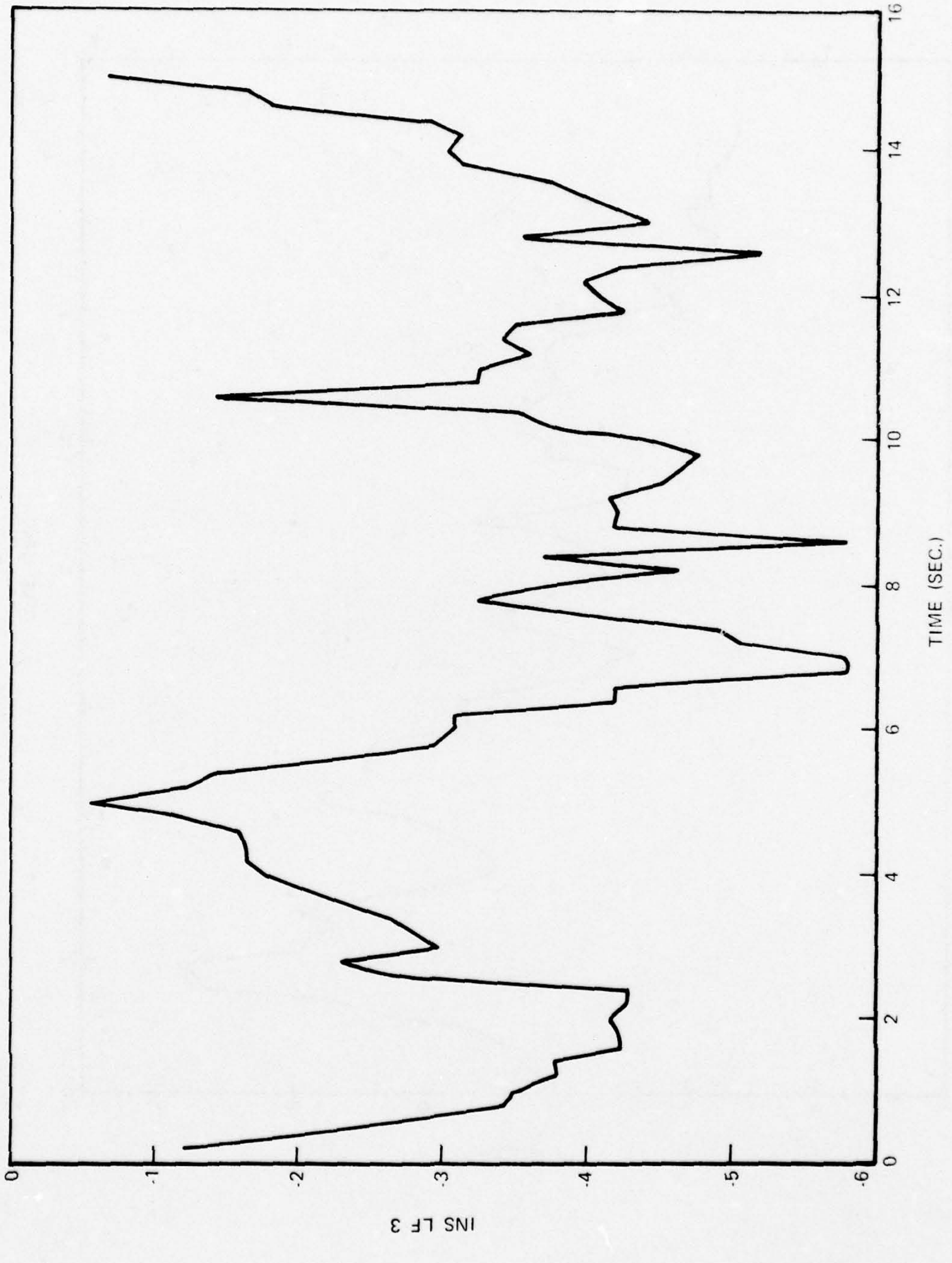


Figure 8c. No Failures; Level Flight

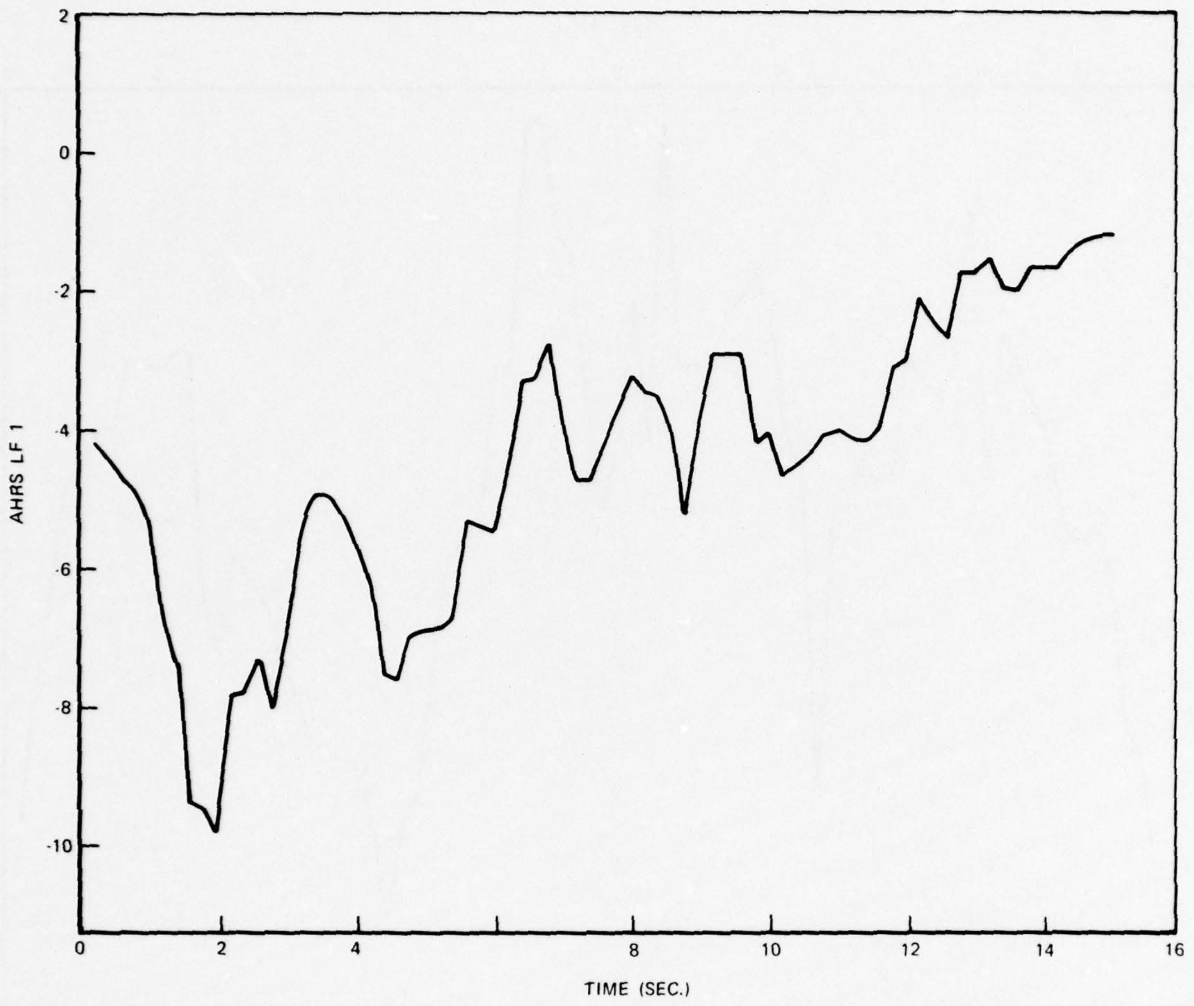


Figure 8d. No Failures; Level Flight

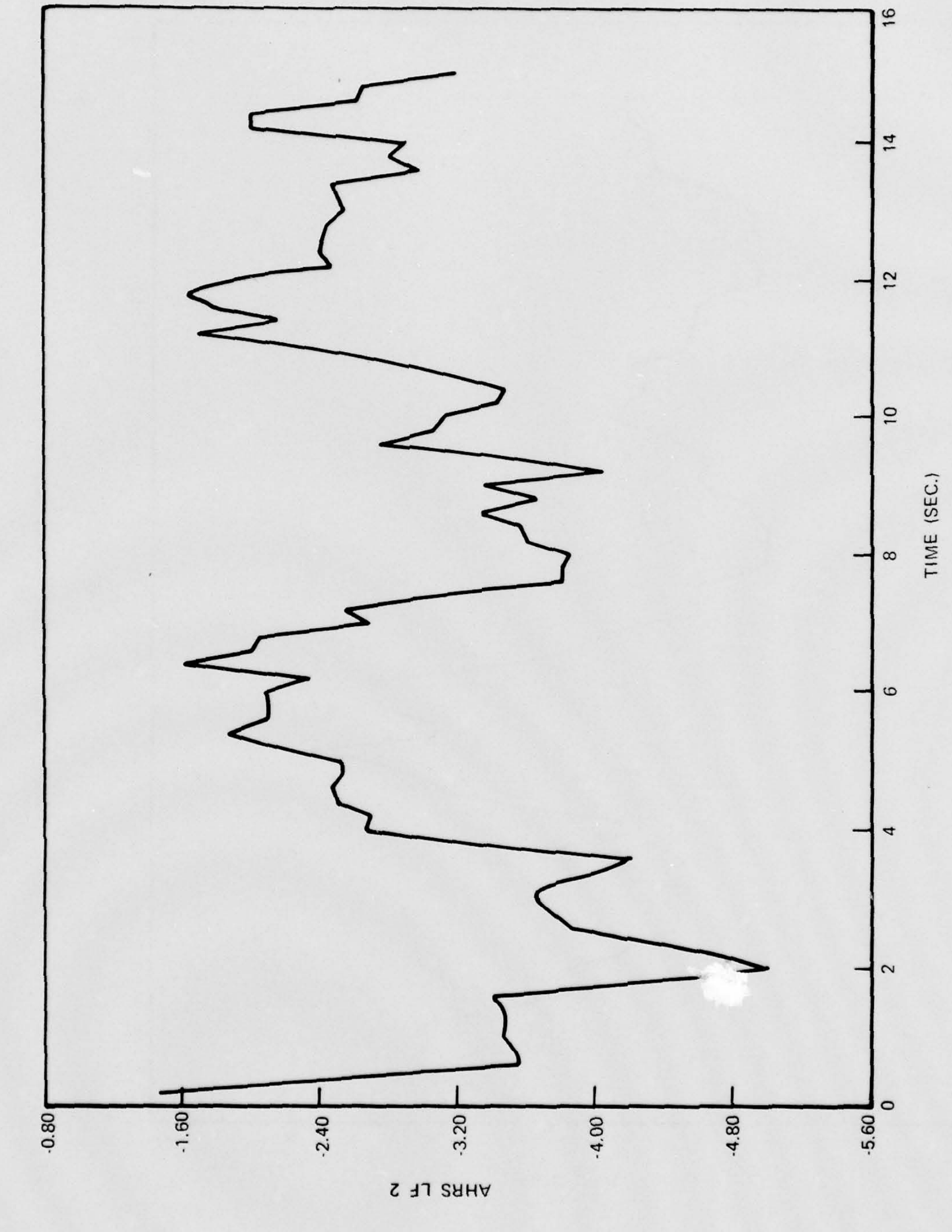


Figure 8e. No Failures; Level Flight

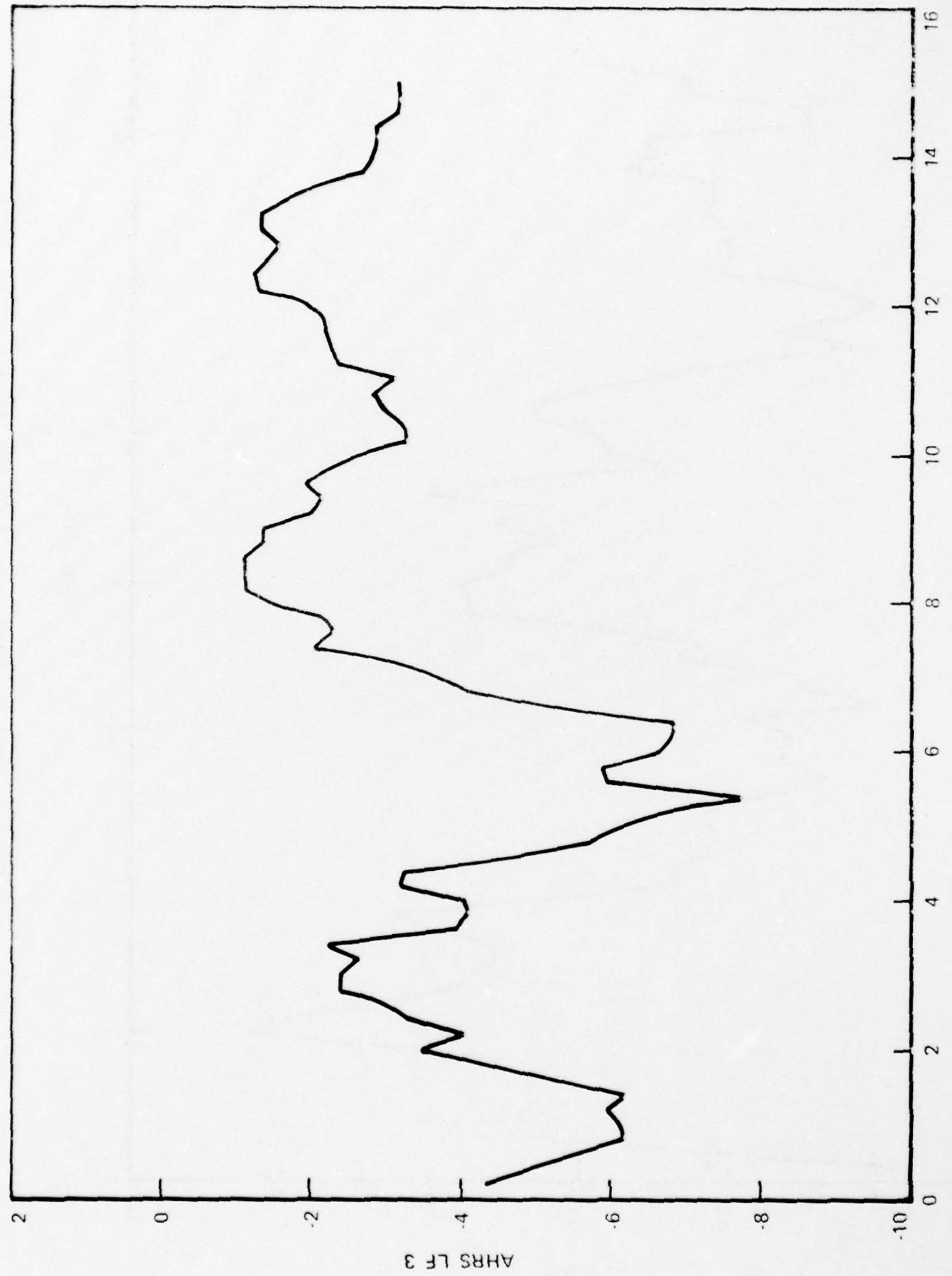


Figure 8f. No Failures; Level Flight

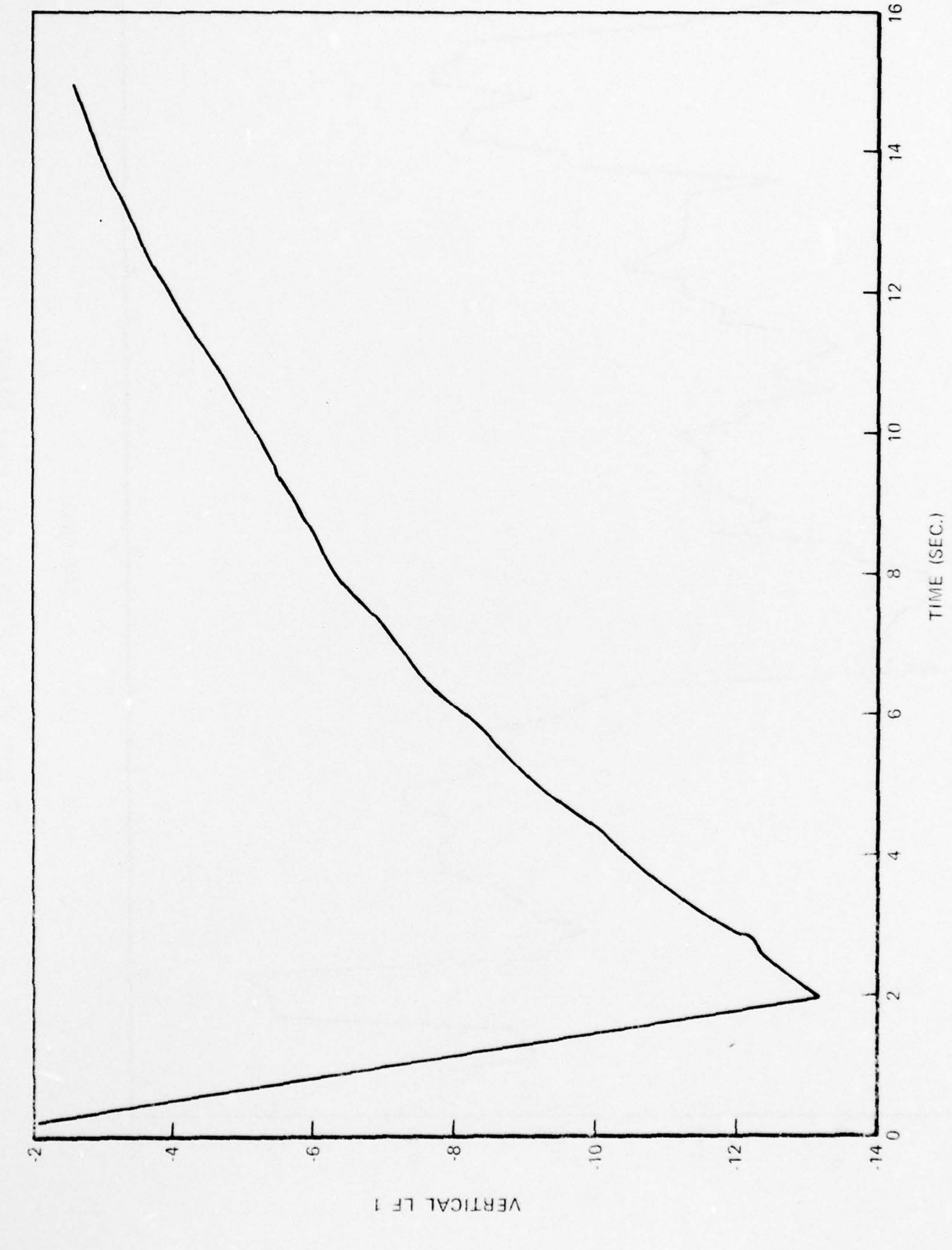


Figure 8g. No Failures; Level Flight

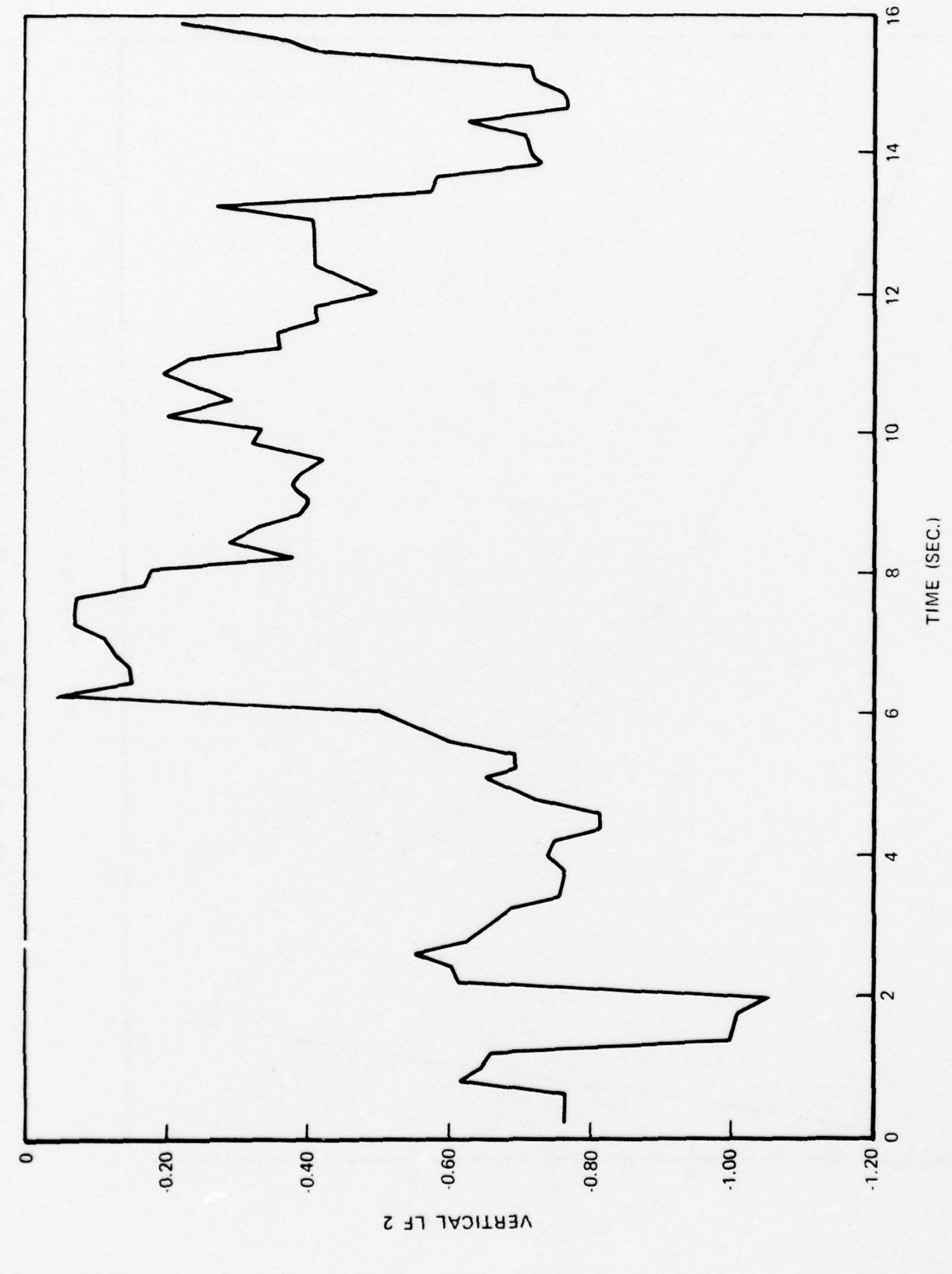


Figure 8h. No Failures; Level Flight

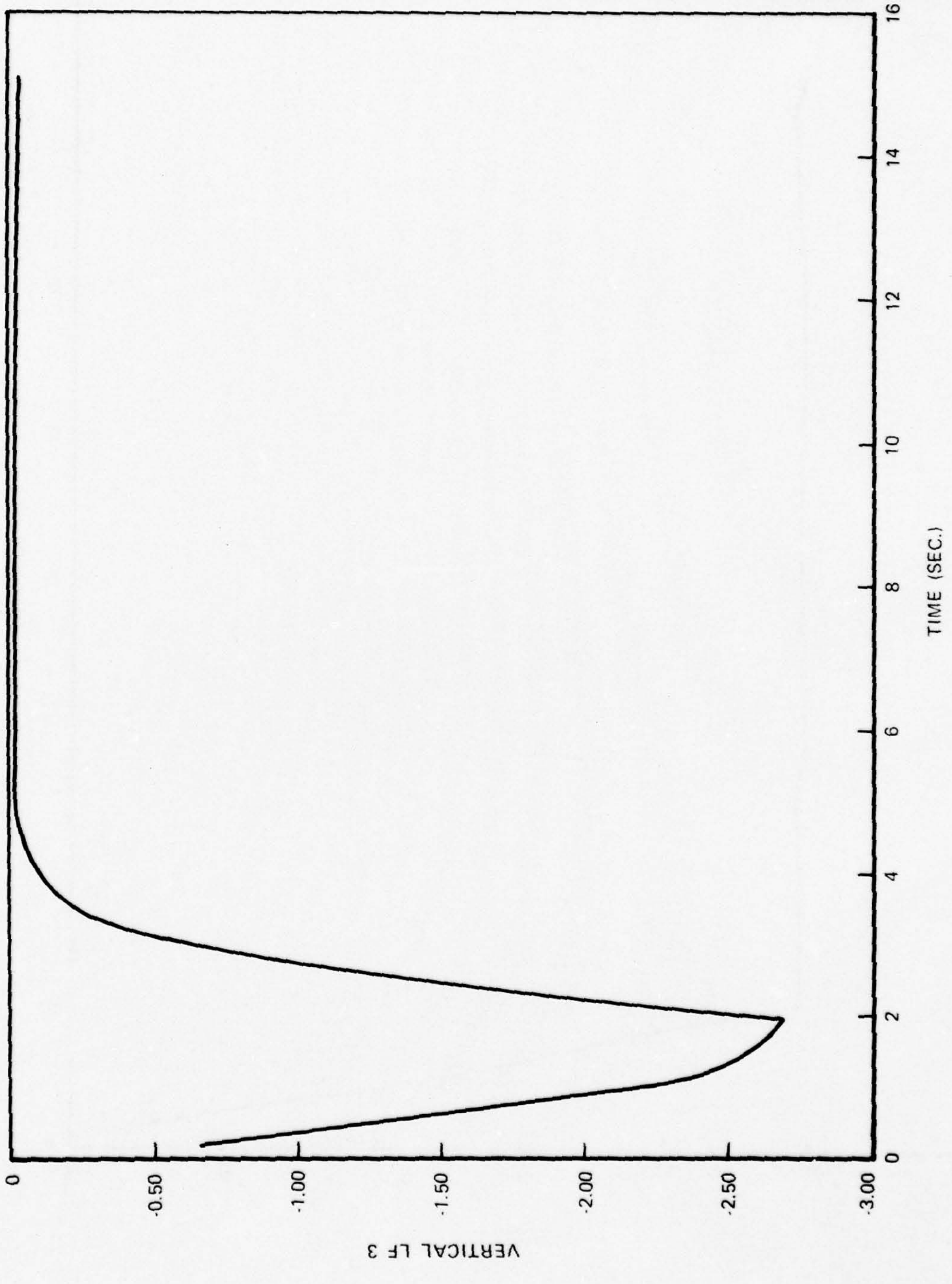


Figure 8i. No Failures; Level Flight

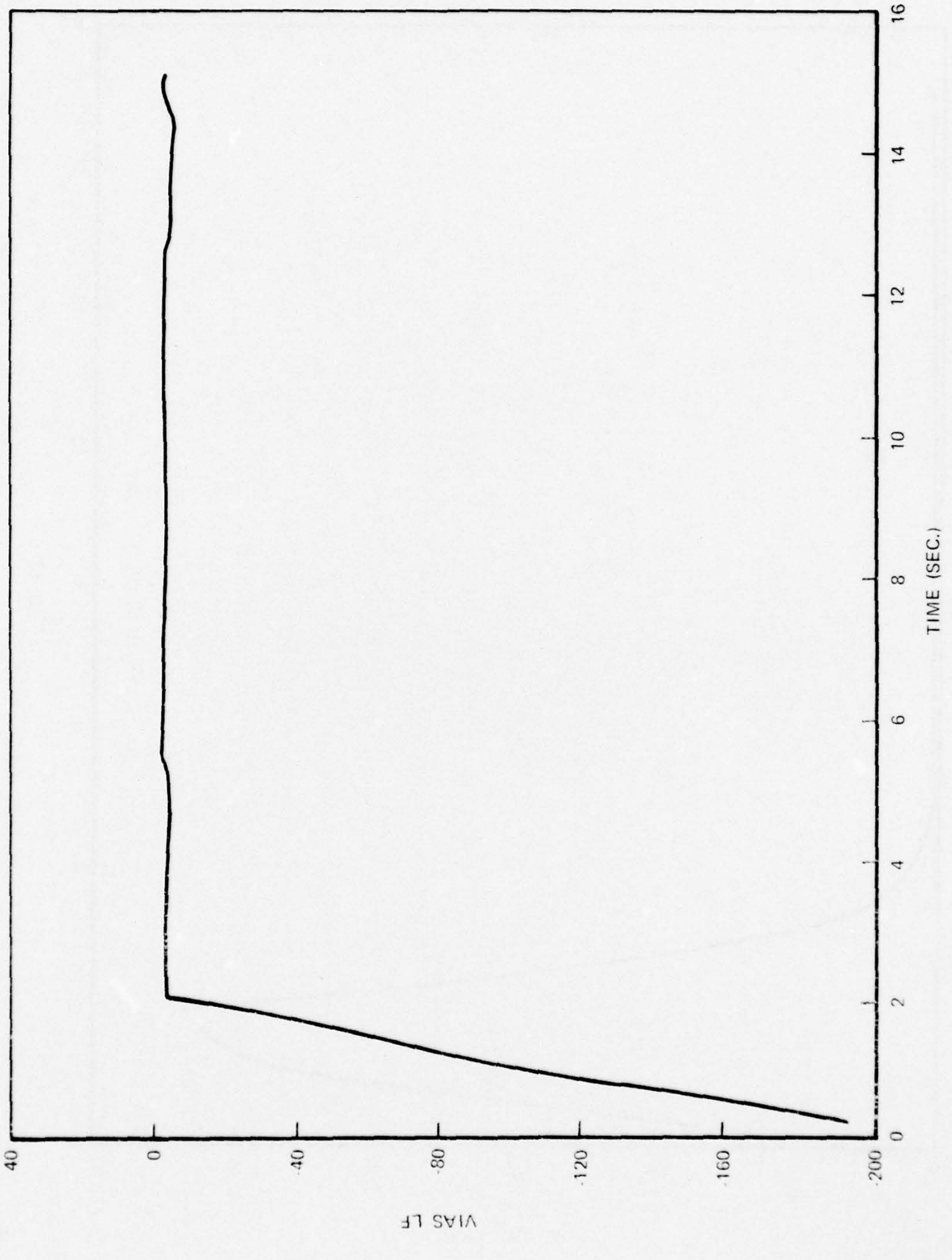


Figure 8j. No Failures; Level Flight

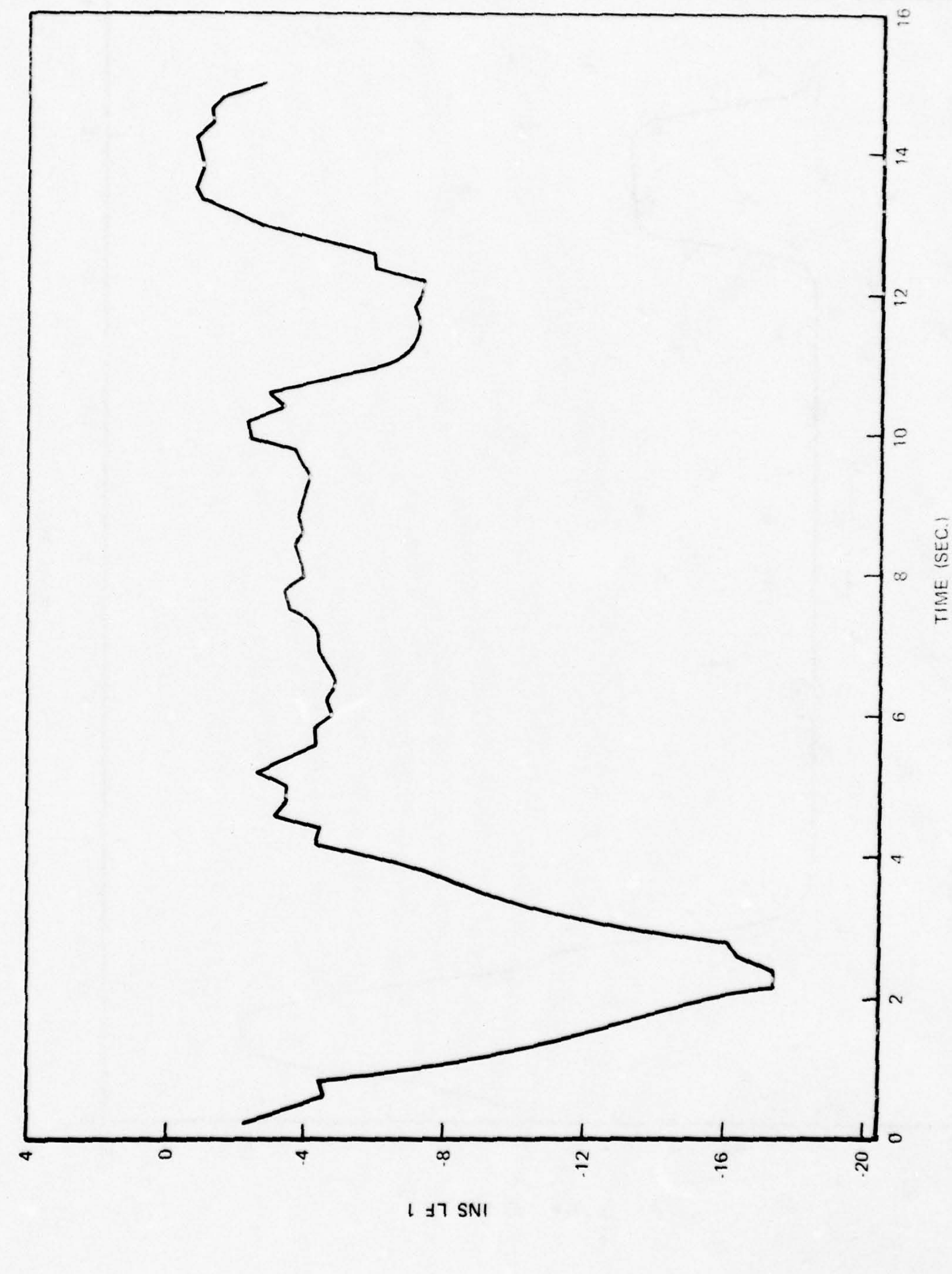


Figure 8k. No Failures; Turn

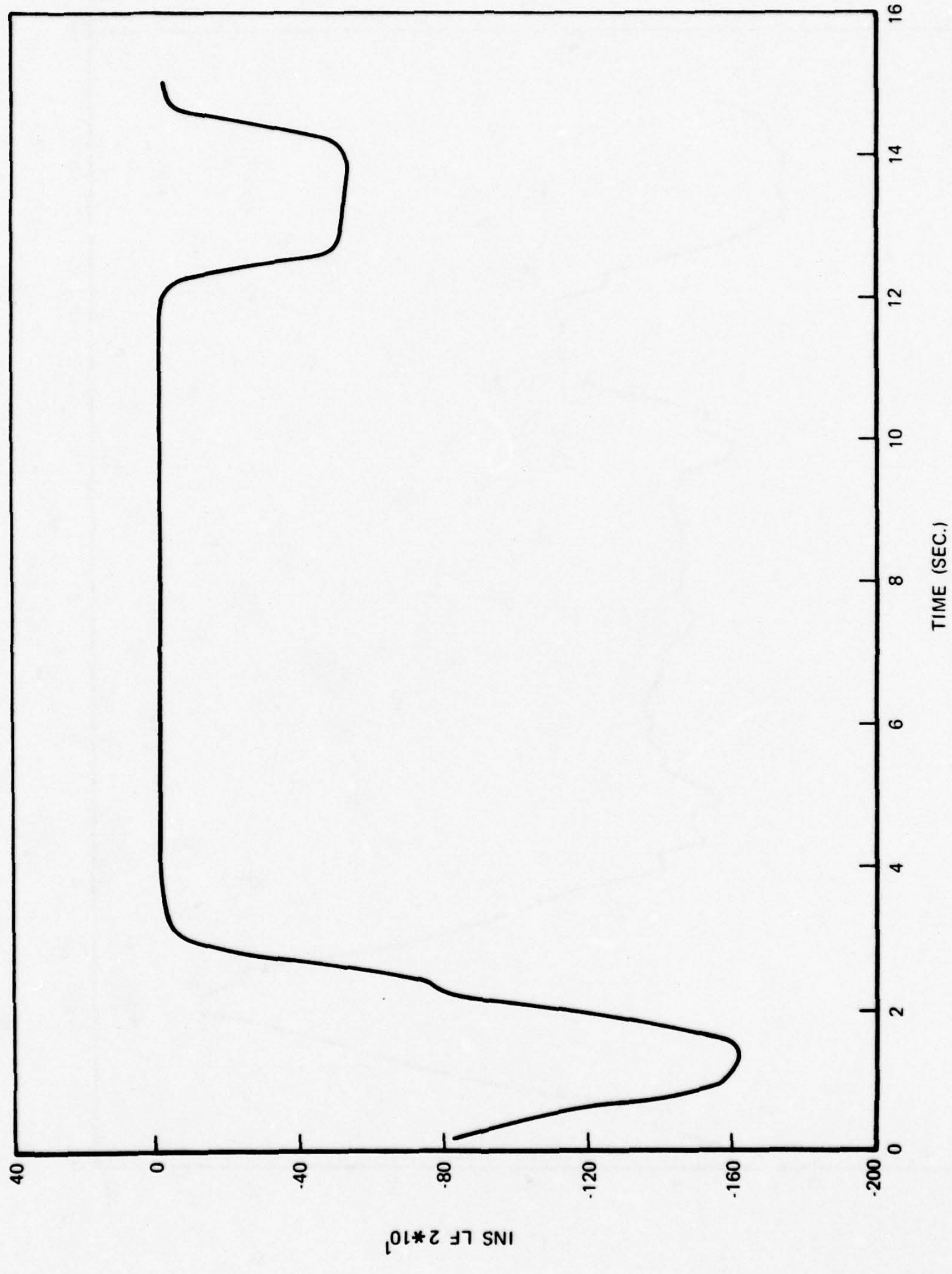


Figure 81. No Failures; Turn

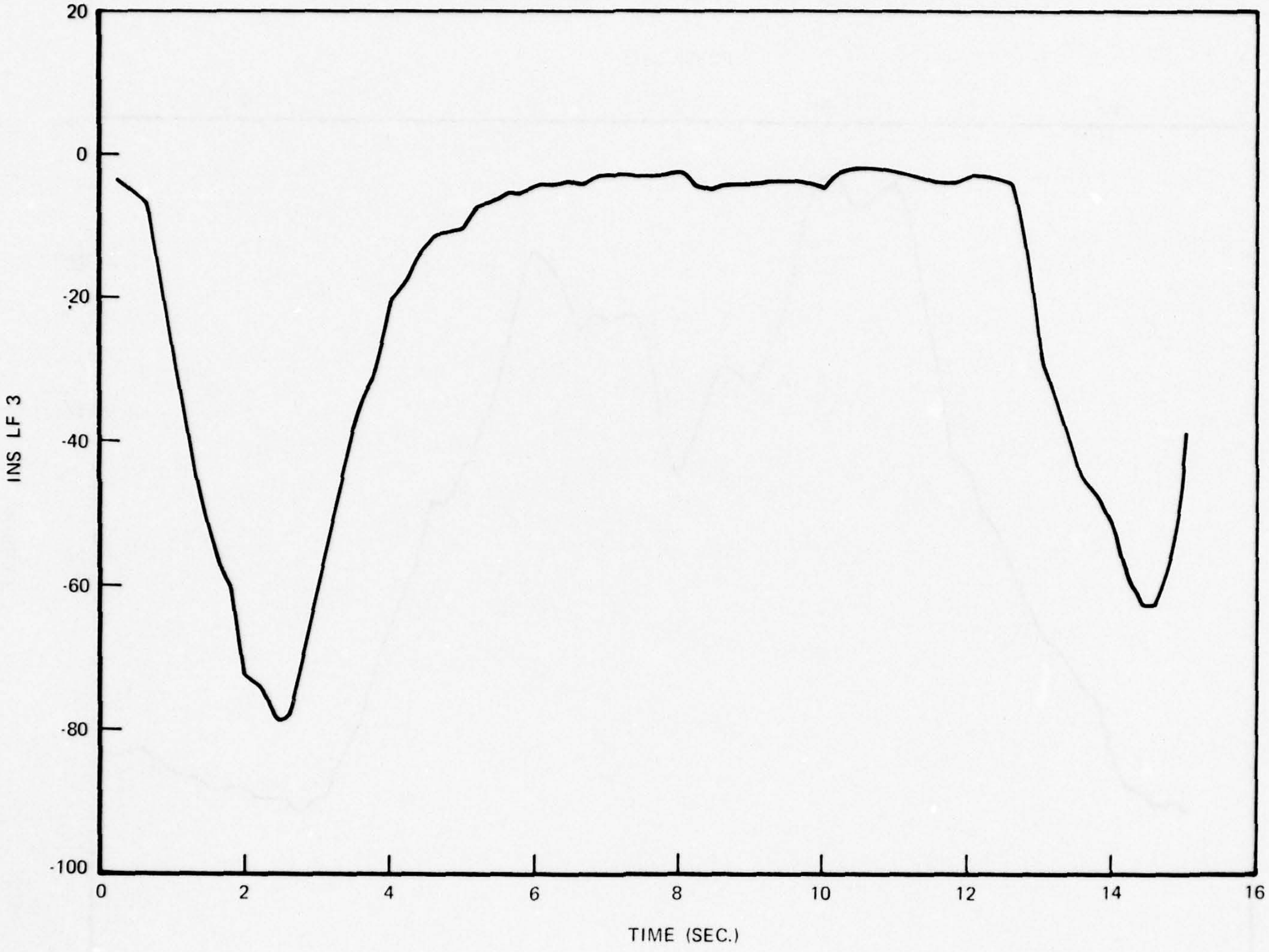


Figure 8m. No Failures; Turn

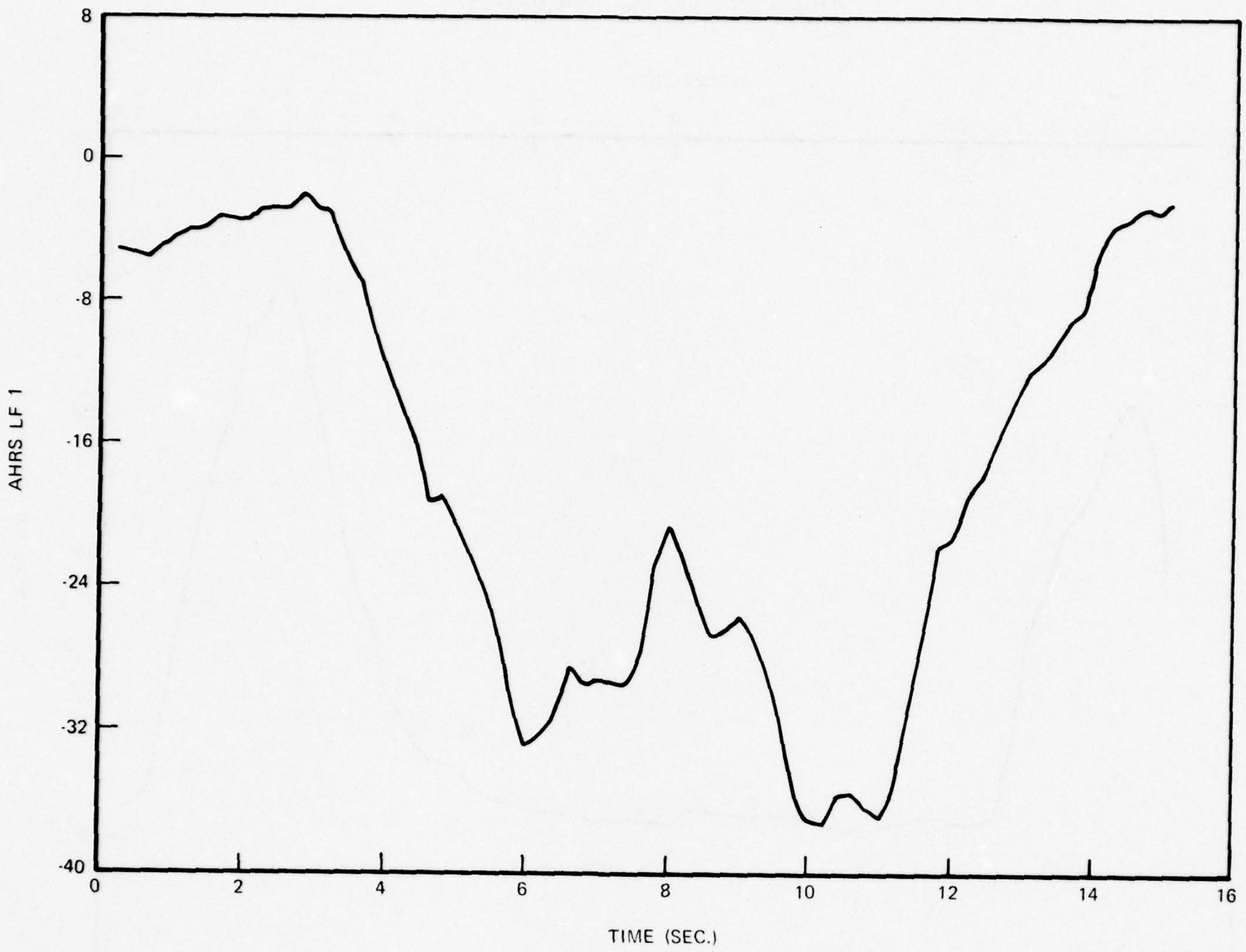


Figure 8n. No Failures; Turn

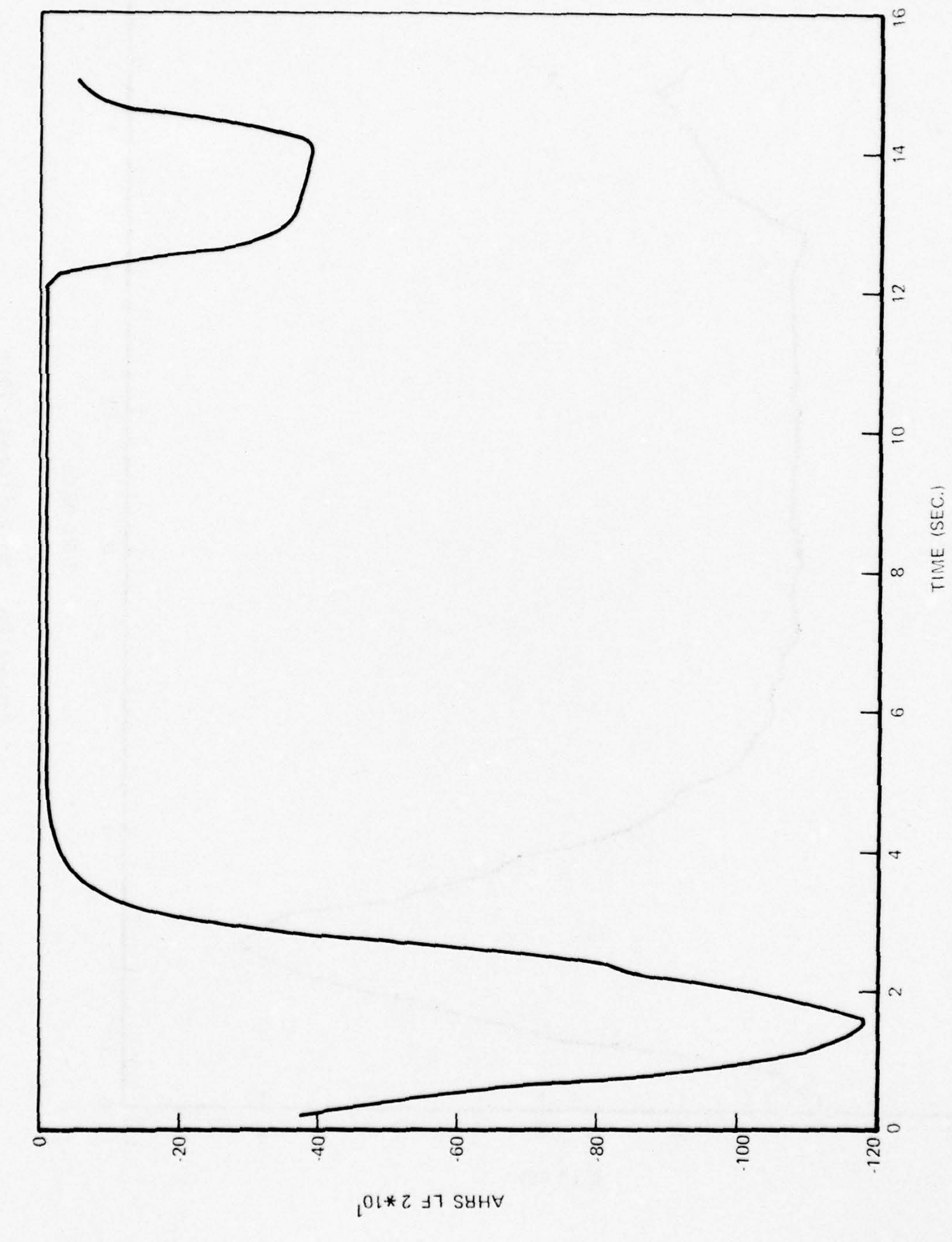
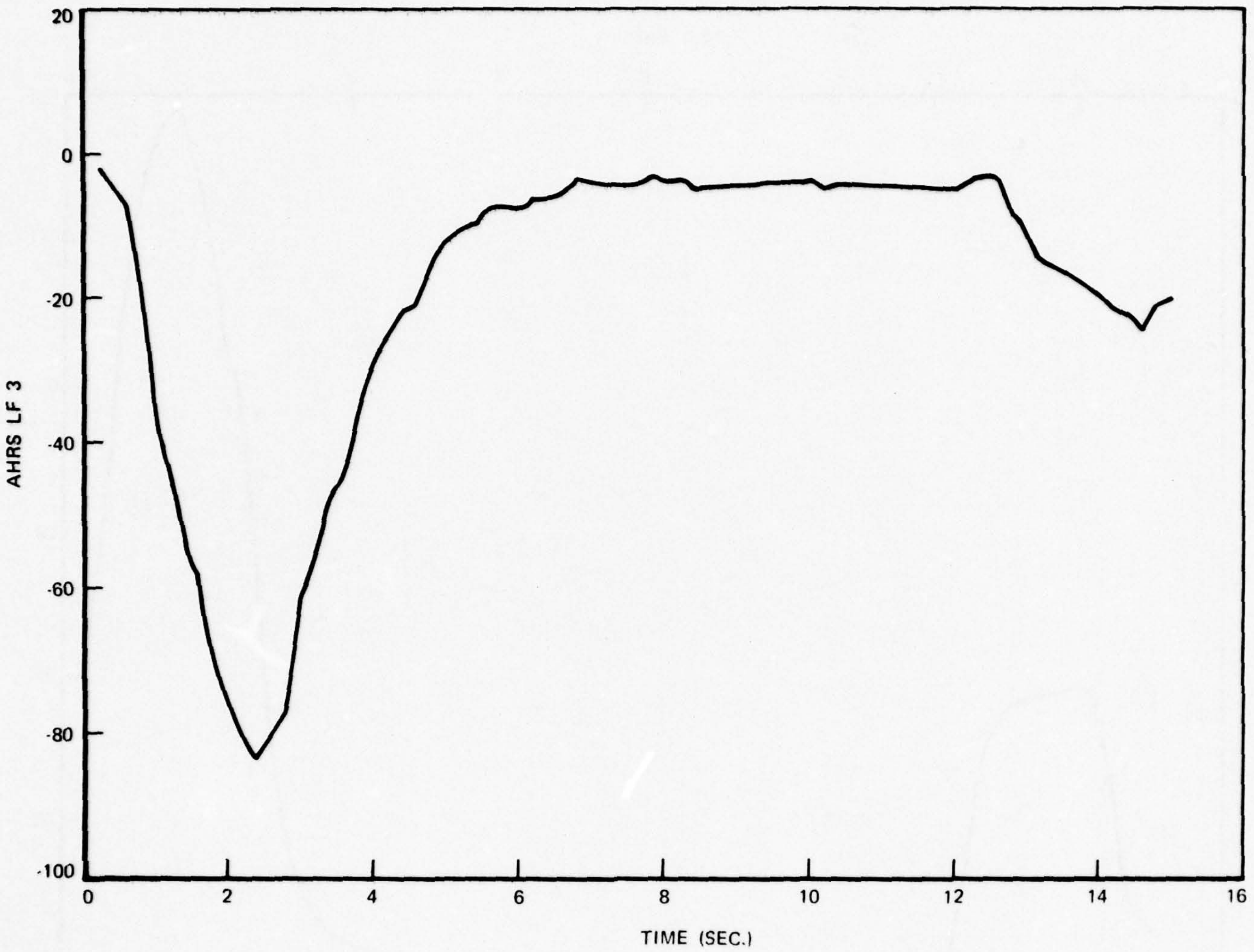


Figure 80. No Failures; Turn



Fiqure 8p. No Failures; Turn

TABLE V

LARGEST LIKELIHOOD FUNCTION VALUES WITH NO FAILURES

LIKELIHOOD FUNCTION	LEVEL	TURN	DESCENT
INS 1	- 4.2	- 20.4	- 6.9
INS 2	- 5.7	-1650	- 11.9
INS 3	- 5.8	- 96	- 5.6
AHRS 1	- 9.0	- 36	- 15.7
AHRS 2	- 5.1	-1185	- 8.6
AHRS 3	- 11.1	- 145	- 6.1
VERT 1	- 13.3	- 14.5	- 14.0
VERT 2	- 1.1	- 5.2	- 4.7
VERT 3	- 2.8	- 2.7	- 2.5
VIAS	-230	- 430	-195

than the -230, -430, and -195, respectively, in the table. Another, less pronounced, effect was that the VERT 2 (independent vertical velocity) achieved maximums of -1.1, -1.5, and -1.8 in the level, turn, and descent phases with no biases simulated, as compared with the -11.1, -5.2, and -4.7 listed in the table. Other than these cases, the biases had only a marginal effect on the likelihood function magnitudes or dynamic characteristics.

The variation of the roll channel likelihood functions of both attitude filters under the influence of rapid roll rates in the turn is especially noteworthy. A variation of greater than two orders of magnitude in these likelihood functions is exhibited in both Figure 8 and Table V. By far, this is the greatest sensitivity of the various likelihood functions to aircraft maneuvering, and therefore it has received intense attention in this effort.

3. FAILURE DETECTION CAPABILITY

This section will describe the results of Monte Carlo runs to verify and improve the performance reported in Tables I and II of Section I. To provide a means of direct comparison, the data will be presented in the same order as in those two tables. Where significant improvement was required and/or achieved, a more detailed accounting of these results will be made.

3.1 SUDDEN FAILURES WITH SUDDEN EFFECTS

First, the study of sudden failures with sudden effects, the content of Table I, will be presented.

Massive leak in static line: During the turn segment, static line leaks of varying degrees were simulated, with such a failure being detected by the altitude (VERT 1; e_7), vertical velocity (VERT 3; e_9), and indicated airspeed (VIAS; e_{11}) likelihood functions surpassing their thresholds. The performance of Table I was achieved, with the smallest leak simulation being within the region such that in-tolerance instrument biases precluded detection in many runs. With more massive leaks (i.e., adding more than 50 to the simulated sensor bias), detection was possible and occurred more rapidly with increasing leak magnitude. The "higher sensitivity" to vertical velocity than altitude exhibited itself in the vertical velocity likelihood function rapidly attaining a value in excess of normal and remaining there, whereas the altitude likelihood function grew more slowly, but continued such growth to surpass the threshold by a greater percentage eventually. Thus, as in Table I, the vertical velocity was signalled in error on the first iteration of the algorithm after the leak was simulated, while the altitude failure required four iterations. Note that, due to simulation errors in the previous work, the indicated airspeed was not affected by static line errors. For the strongest leak simulated (adding 500 to the sensor bias), the altitude likelihood function exceeded its normal peak value by about 7, the vertical velocity by about 4.5, and the indicated airspeed by about 1100.

Some difficulty with the airspeed (VIAS) likelihood function growing large was experienced, but this was attributed to other instrument biases. Thus, these were not false alarms caused by the simulated line leak.

Massive leak in pitot line: The performance of Table I was achieved when instrument biases were zeroed, but some difficulty was experienced when other instrument biases were allowed to affect the system. The indicated airspeed (VIAS) likelihood function repeatedly demonstrated such performance.

Excessive noise in static pressure output: The altitude (VERT 1), vertical velocity (VERT 3), and indicated airspeed (VIAS) likelihood functions signalled such a static pressure sensor failure, with similar trends as in Table I. For the largest noise power simulated, VERT 1 attained approximately -31 (as compared to -14.5 as listed in Table V for the greatest value under no-fail conditions), VERT 3 achieved about -28 (compared to -2.7) and VIAS achieved -1500 (compared to -430). This is seen in Figure 9, typical plots of VERT 1 and VERT 2 under conditions of their largest failures.

The VERT 2 likelihood function magnitude grew somewhat, though not as severely as VERT 1 and VERT 3 or enough to surpass threshold. This can be attributed to the vertical filter being degraded by continued use of a failed signal.

Excessive noise in pitot pressure output: As indicated in Table I, for large enough noise corruption, a failure is declared by the airspeed (VIAS) likelihood function exceeding threshold. For $\sigma_p^* = 600$, the value grows to about -750, and for $\sigma_p^* = 1200$ it grows to -1800 then oscillates back to -600, both of these being considerably beyond the -430 value in

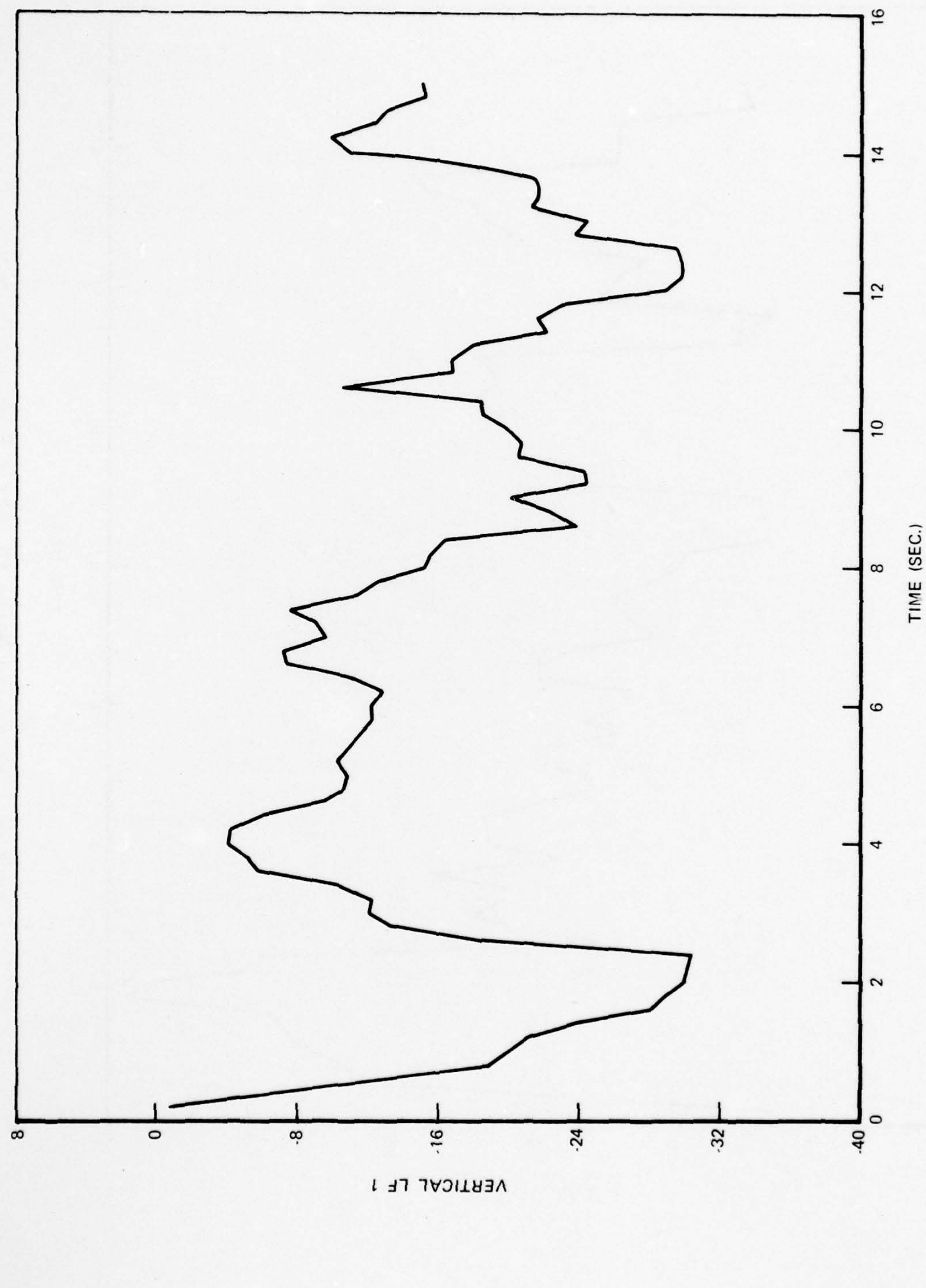


Figure 9a. Excessive Noise in Static Pressure Output

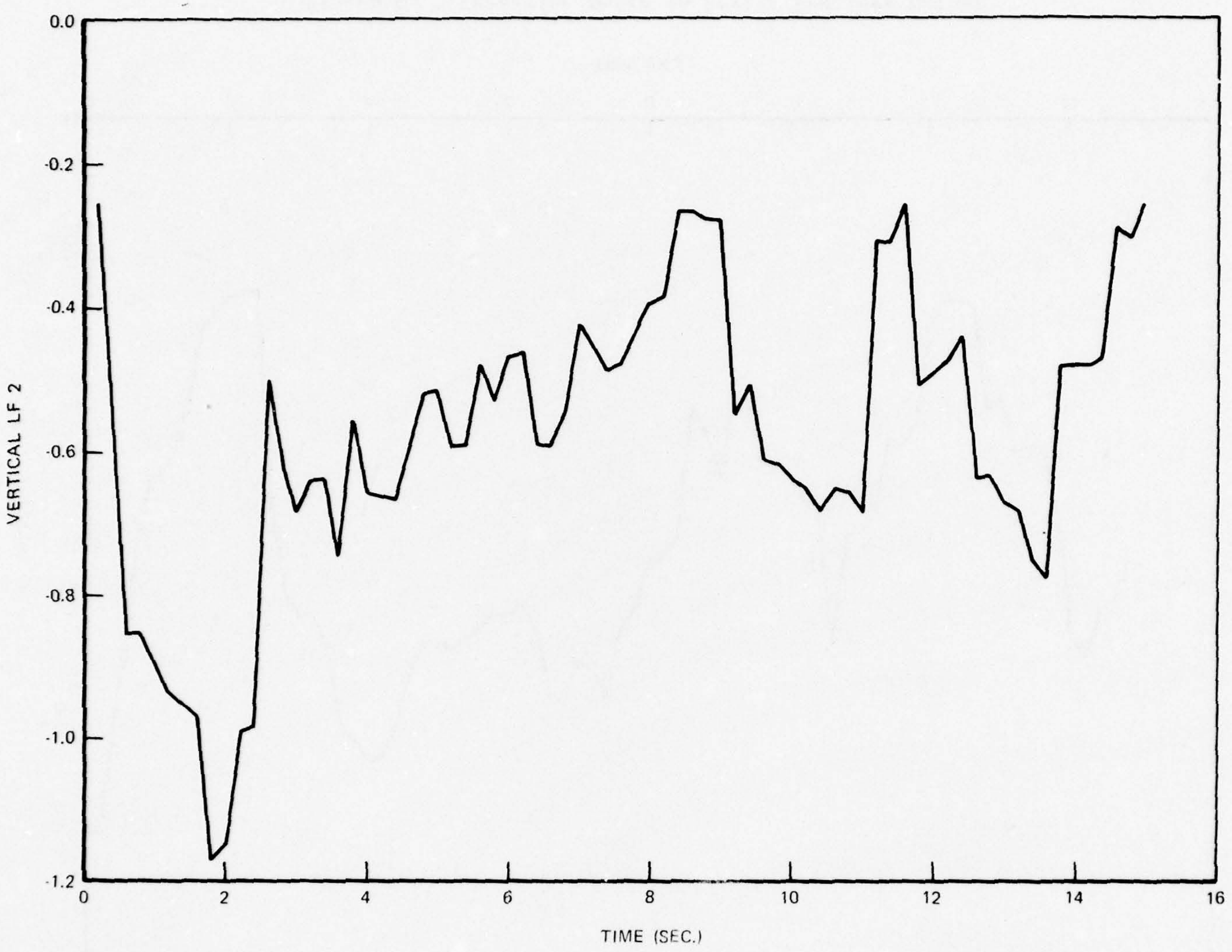


Figure 9b. Excessive Noise in Static Pressure Output

Table V under no-fail conditions, and especially beyond -20 under no-fail conditions and no instrument biases (as mentioned in Section IV.2). Figure 10 portrays the VIAS likelihood function for a run with σ_p^* = 1200.

Tachometer failure: When the tachometer failed in descent, the vertical velocity likelihood function signalled a failure, usually after about one second, with the function approximately doubling its "normal operation maximum" value after two seconds.

Bent angle-of-attack vane: Similar to the indicated performance in Table I, for the low values of additional bias used to simulate the bent angle-of-attack vane, the results did not consistently put the VERT 2 likelihood function level beyond the value of -5.2 shown in Table V. However, when 0.06 was added to b_α, the value grew to -6 in approximately two seconds, and remained at that level. For an additional bias of 0.12, the level grew to -36 in about two seconds, as shown in Figure 11a. Unlike the results in Table I, the indicated airspeed likelihood function did not in general grow beyond the -430 level of Table V, although it did usually grow beyond the zero bias level of -20.

As might be expected, the other estimates in the vertical filter were degraded somewhat due to incorporation of faulty data, and so VERT 1 and VERT 3 did in fact grow in magnitude. However, as seen in Figures 11b and 11c, these two likelihood functions underwent growth substantially lower than the corresponding likelihood function. Thus, any potential false alarm from their growth could probably be precluded by removing the failed signal from the filter.

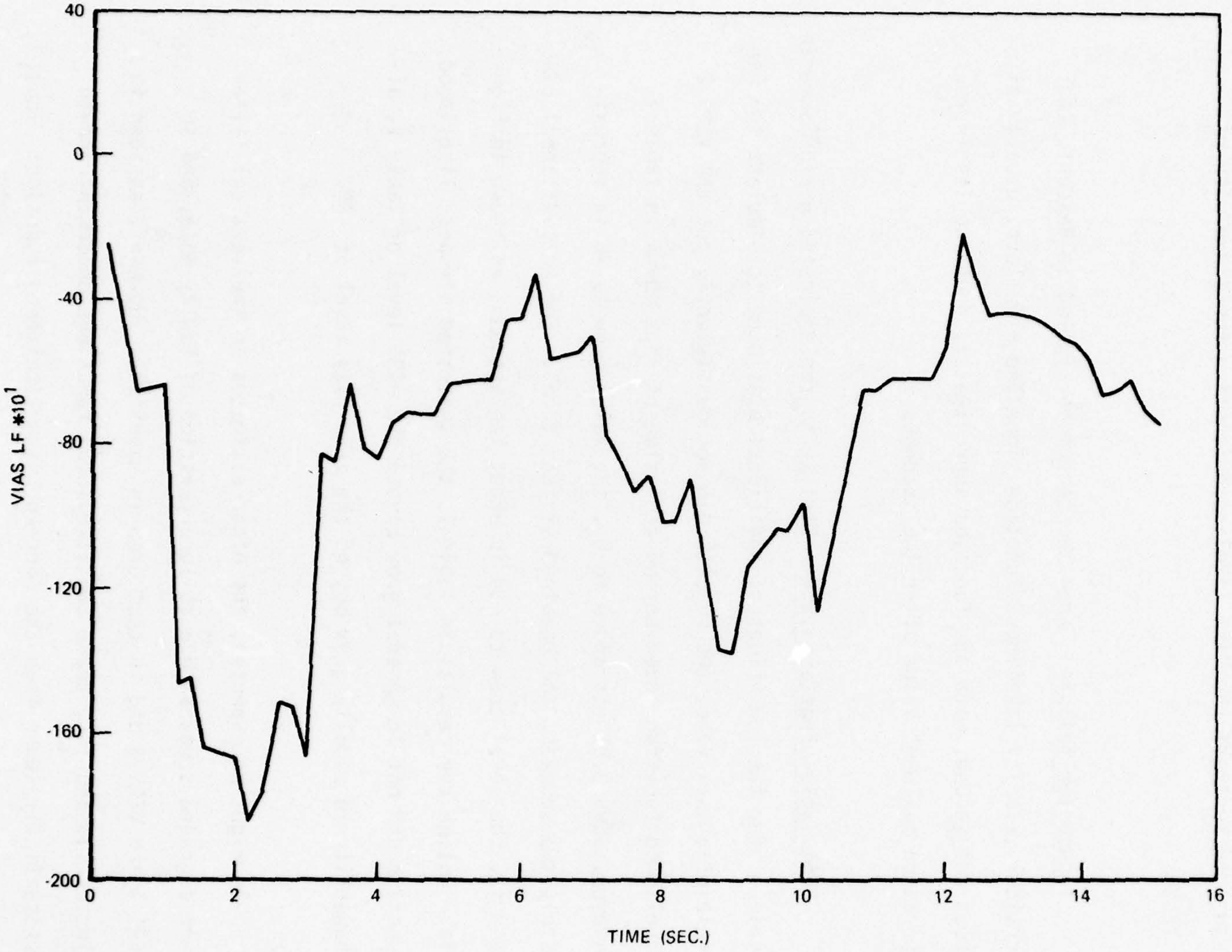


Figure 10. Excessive Noise in Pitot Pressure Output

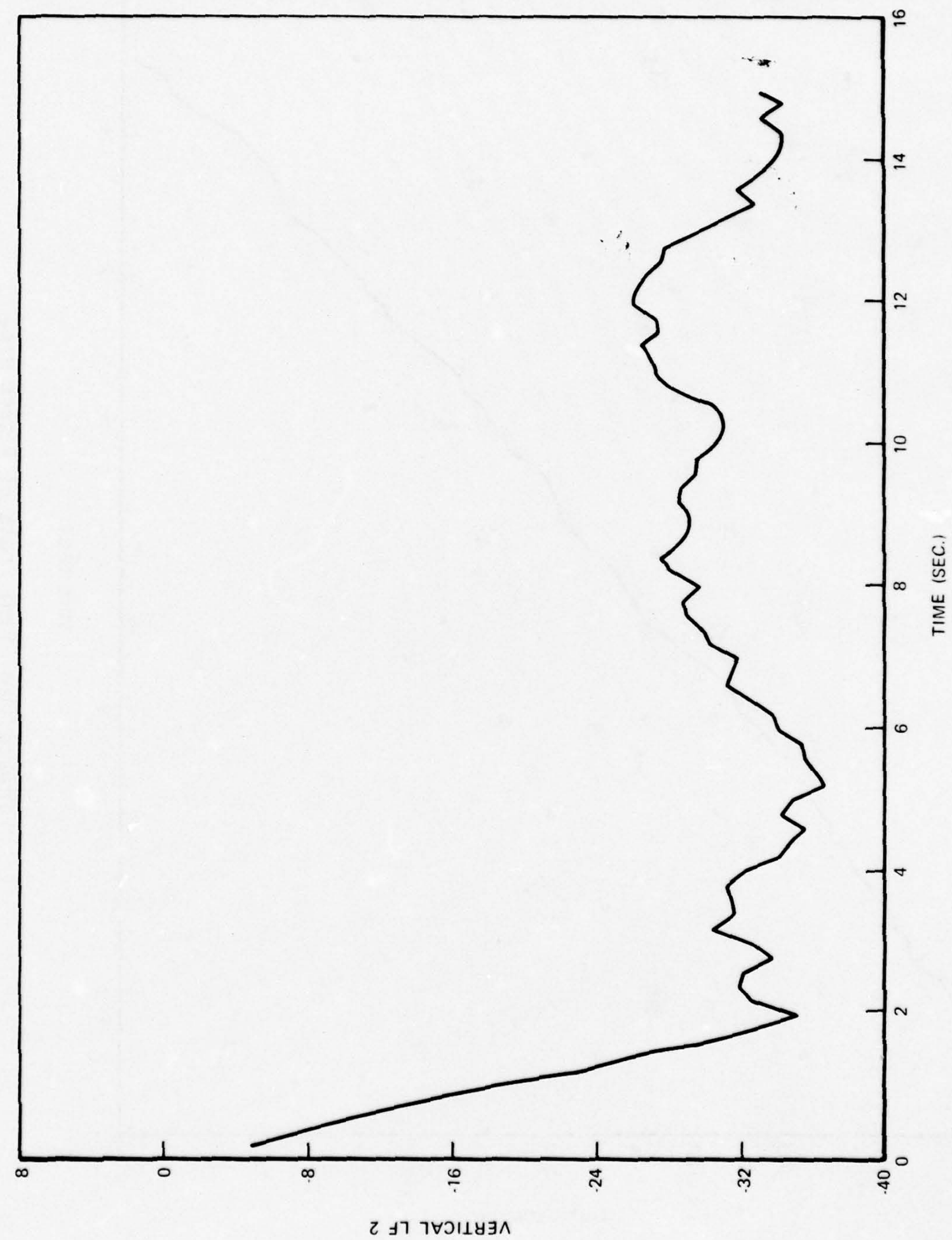


Figure 11a. Bent Angle of Attack Vane

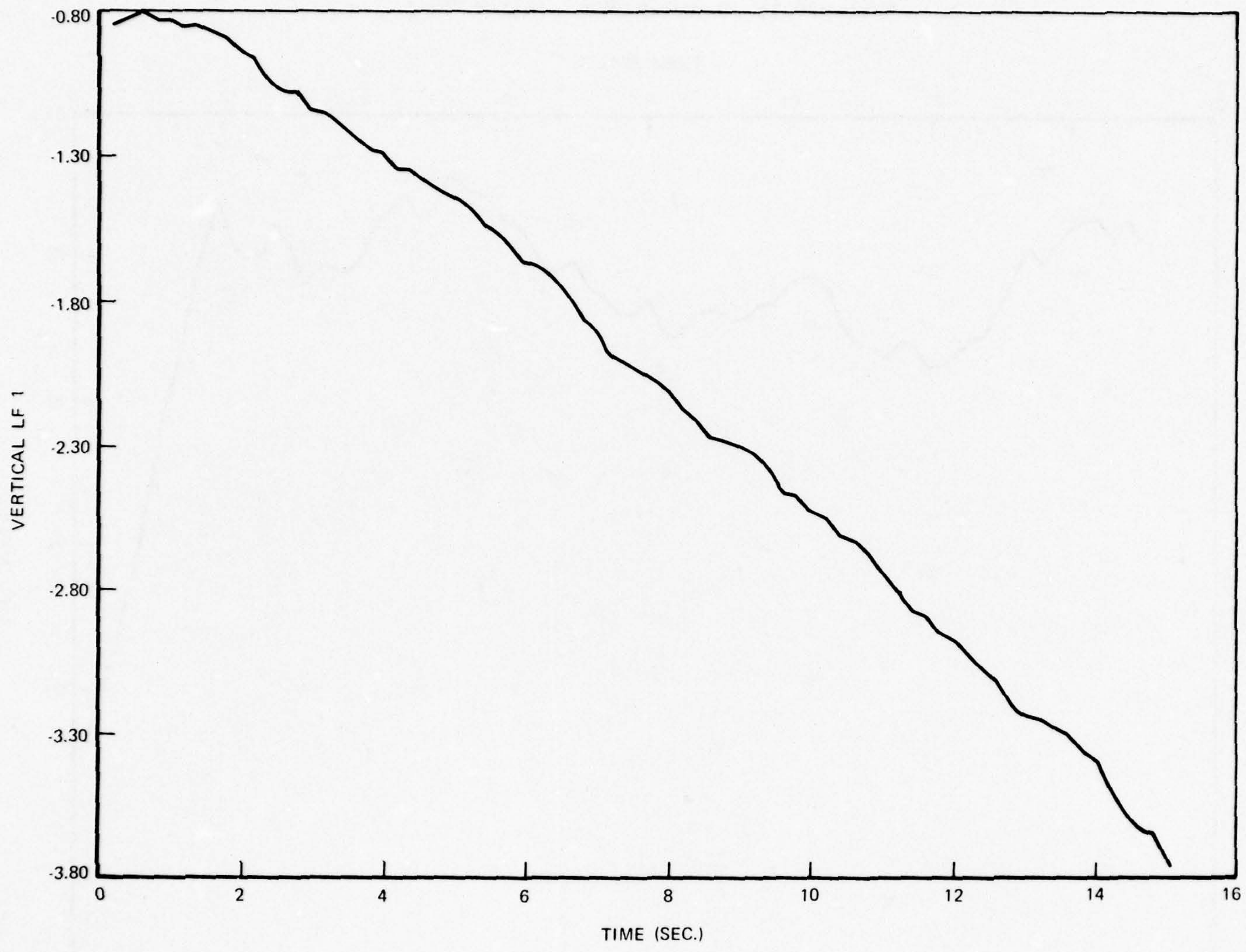


Figure 11b. Bent Angle of Attack Vane

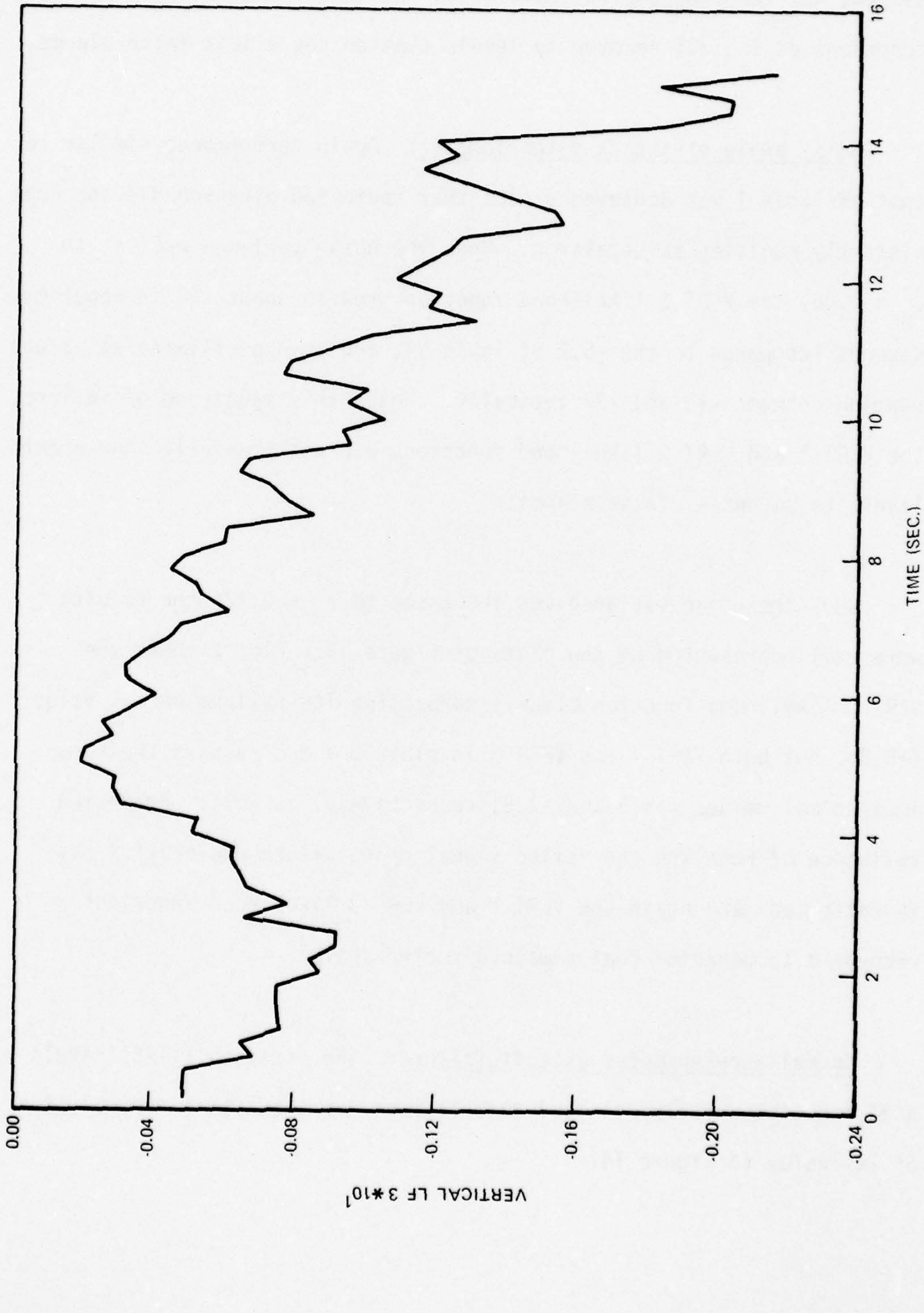


Figure 11c. Bent Angle of Attack Vane

This idea was tested, and Figure 12 presents the confirmation obtained. By removing the failed signal, the VERT 1 and VERT 3 likelihood functions do in fact recover to levels that do not elicit false alarms.

Noisy angle-of-attack potentiometer: Again performance similar to that of Table I was achieved except that indicated airspeed did not consistently register false alarms. When the noise variance was set to $\sigma_\alpha = 0.06$, the VERT 2 likelihood function grew to about -25 in about two seconds (compared to the -5.2 of Table V), and then oscillated at values ranging between -15 and -27 typically. Under this magnitude of failure, the VERT 1 and VERT 3 likelihood functions did not generally show growth levels to potential false alarms.

When the noise variance was increased to $\sigma_\alpha = 0.12$, the results were well represented by the plots of Figure 13. Plot a shows the VERT 2 likelihood function clearly surpassing its maximum normal value (-5.2), but both VERT 1 and VERT 3 in plots b and c surpass their maximum normal values (-4.5 and -2.8, respectively) as well. Again the influence of removing the failed signal upon failure declaration was investigated, and again the VERT 1 and VERT 3 likelihood functions recovered to behavior that remained subthreshold.

Normal accelerometer pickoff failure: The airspeed (VIAS) likelihood function quickly detected this failure, as shown by a typical plot of its value in Figure 14.

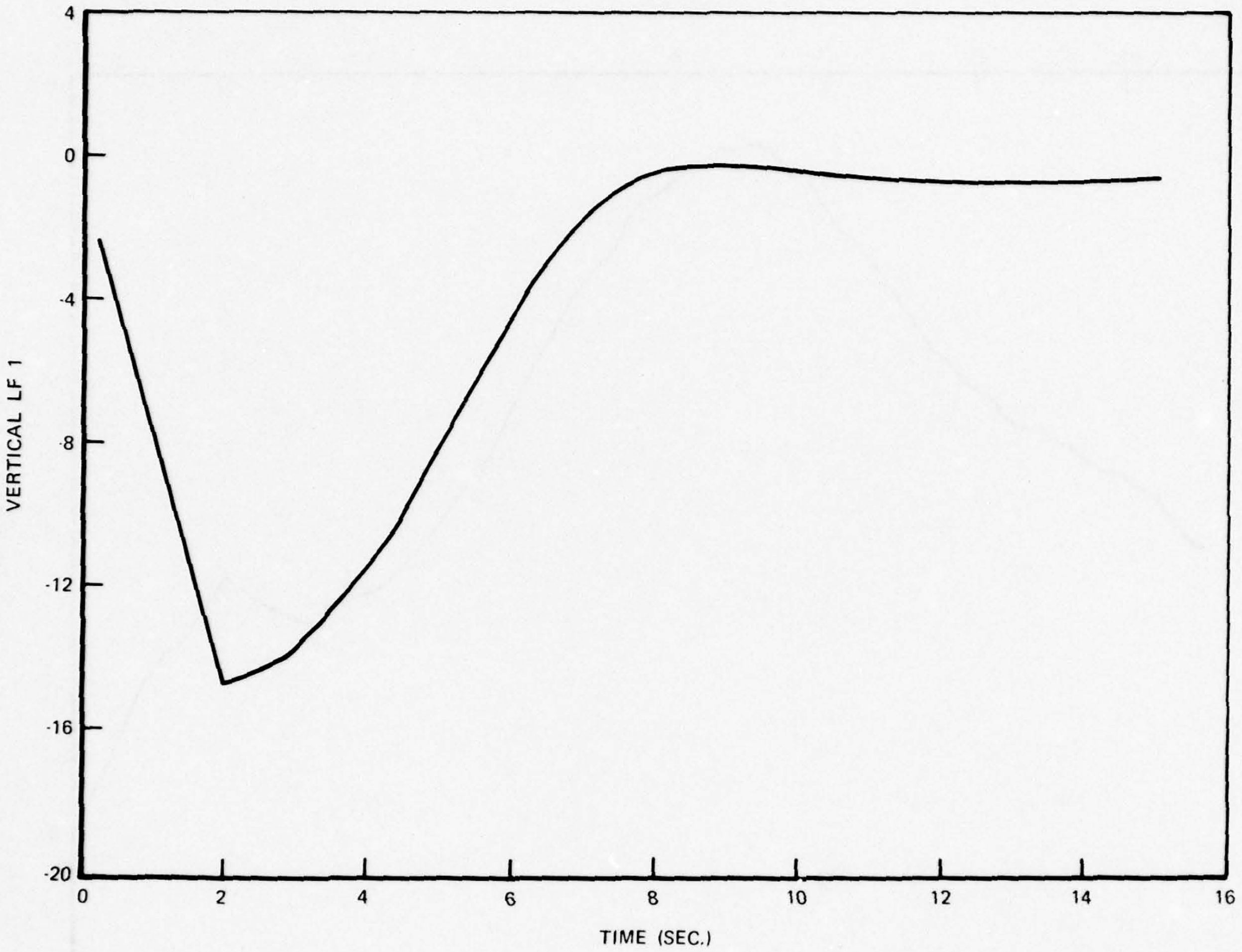


Figure 12a. Recovery of Vertical Likelihood Function 1

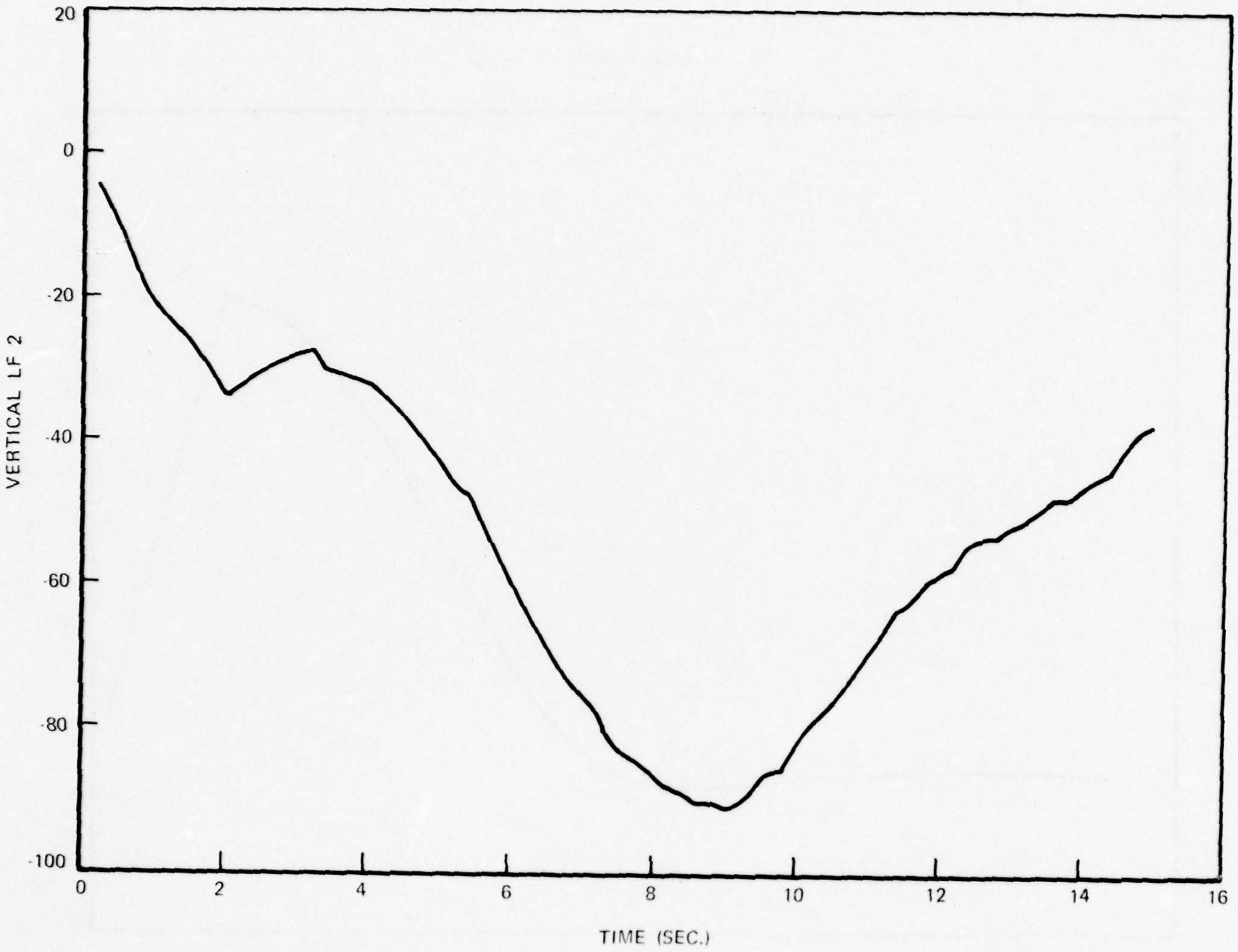


Figure 12b. Recovery of Vertical Likelihood Function 2

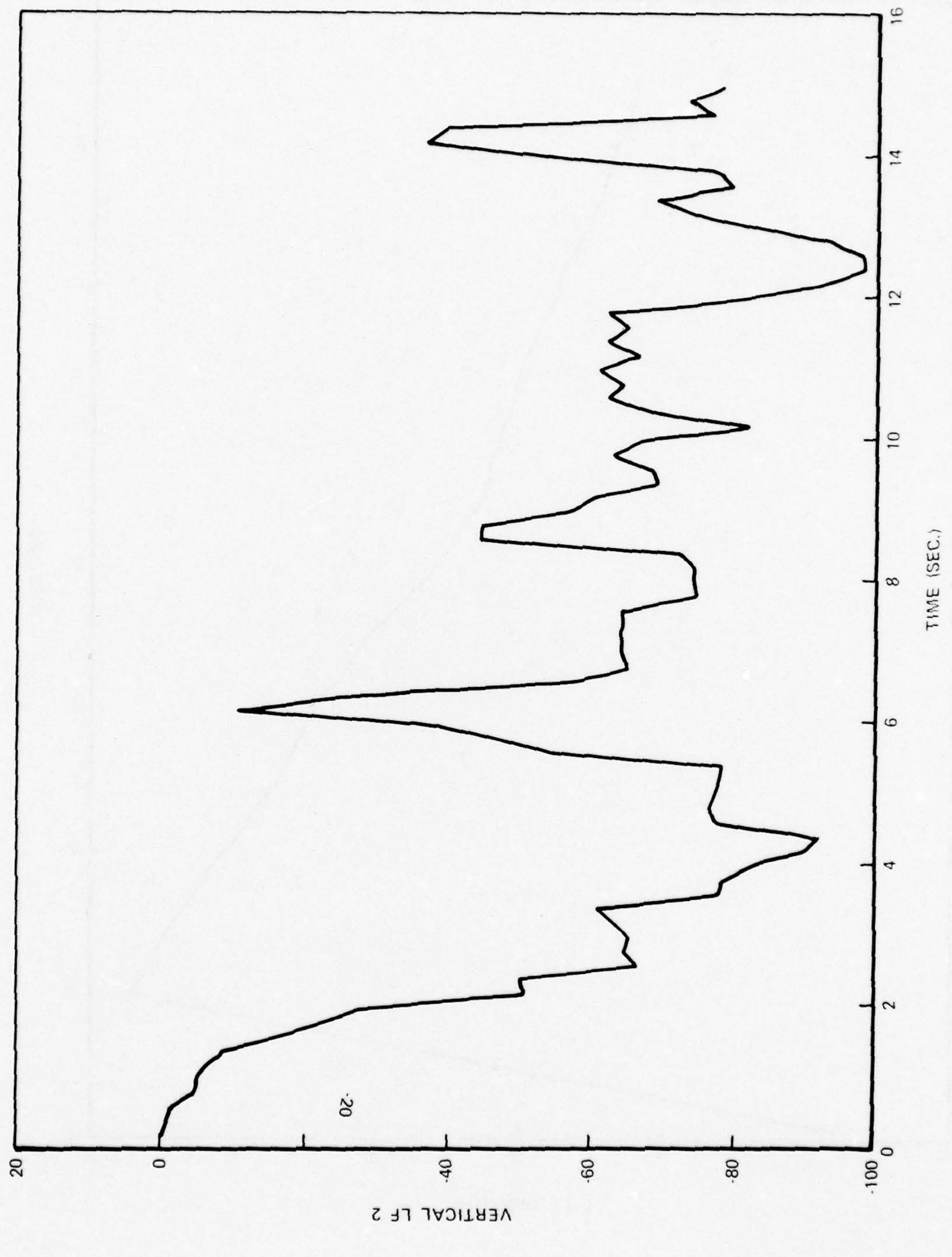


Figure 13a. Noisy Angle-of-Attack Potentiometer

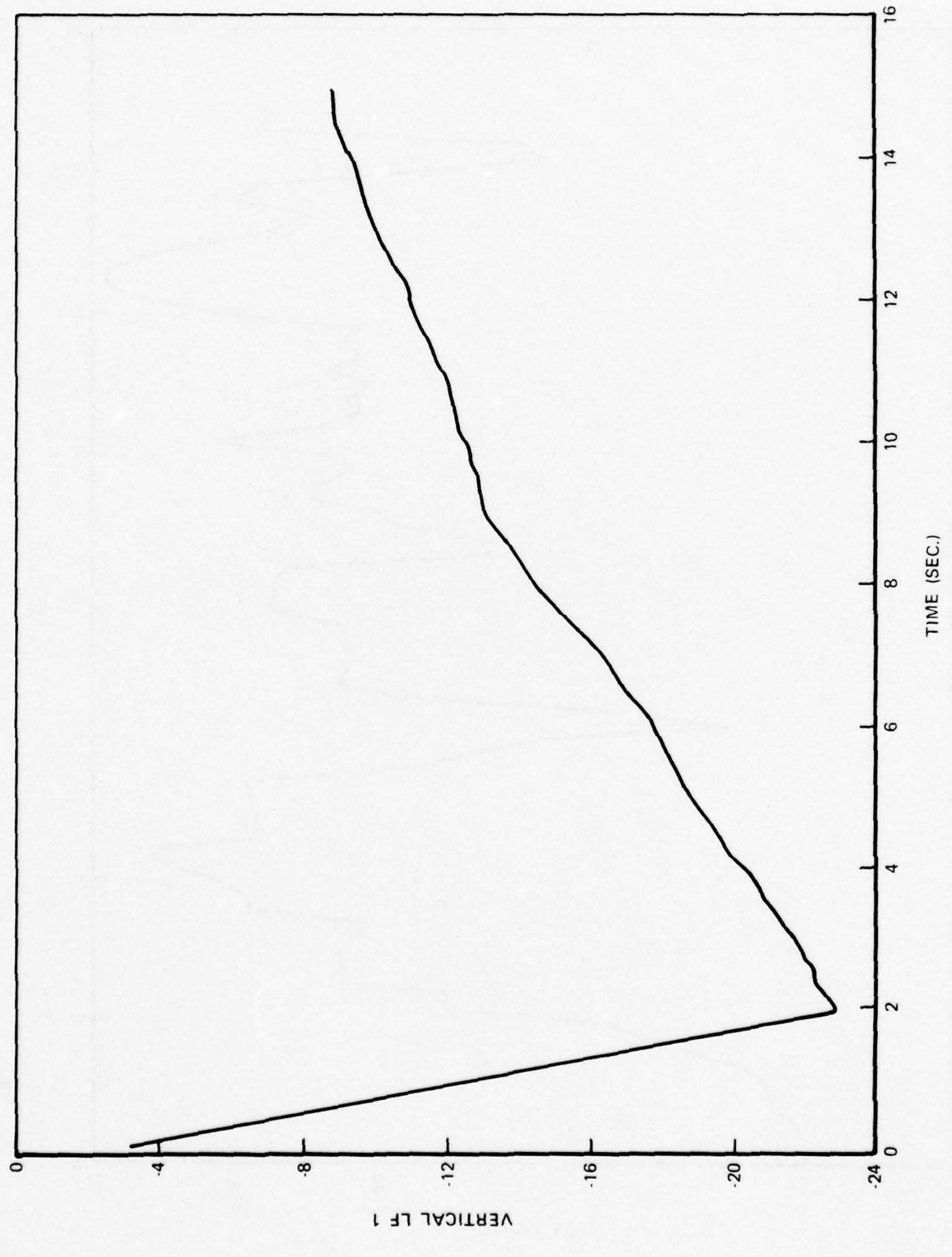


Figure 13b. Noisy Angle-of-Attack Potentiometer

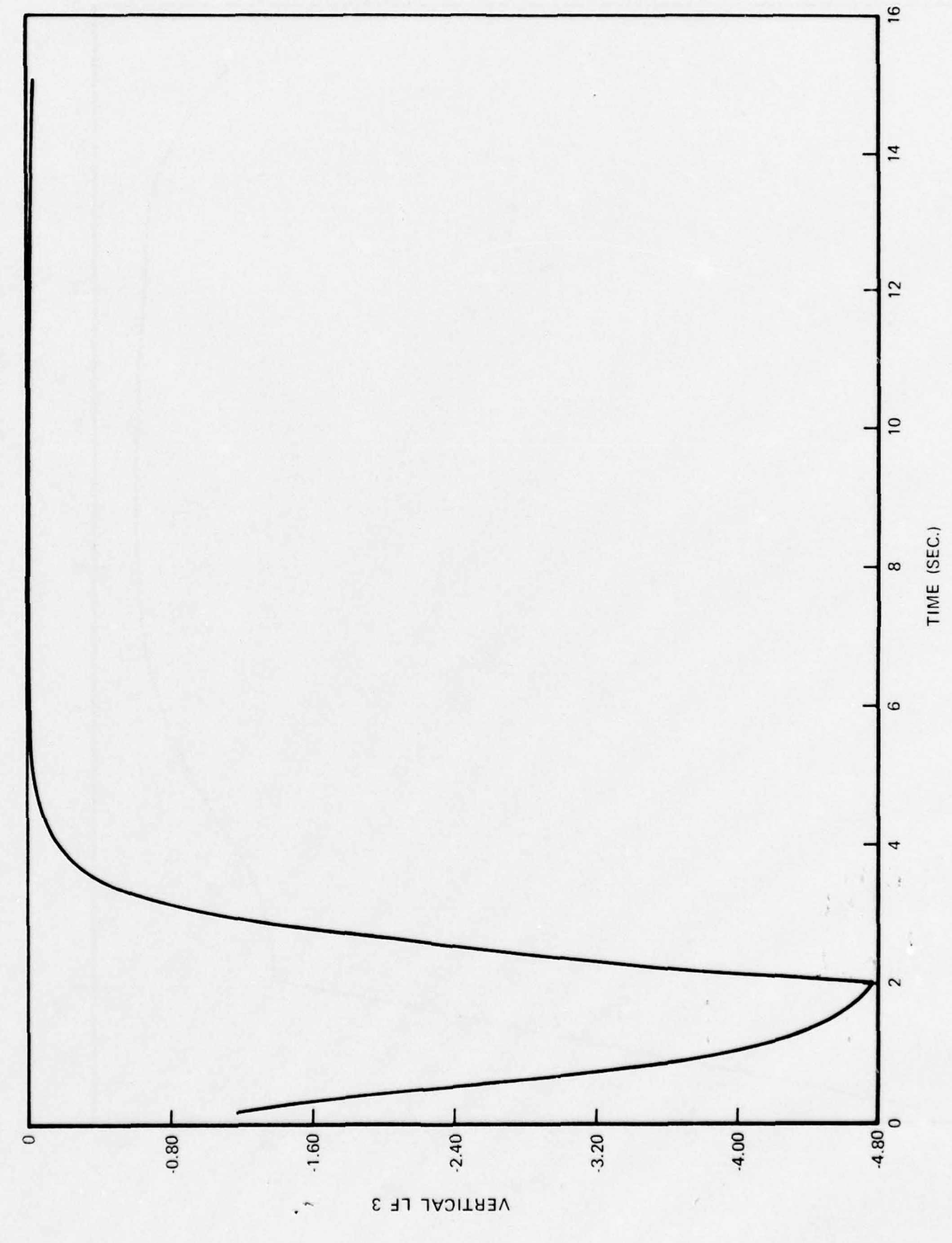


Figure 13c. Noisy Angle-of-Attack Potentiometer

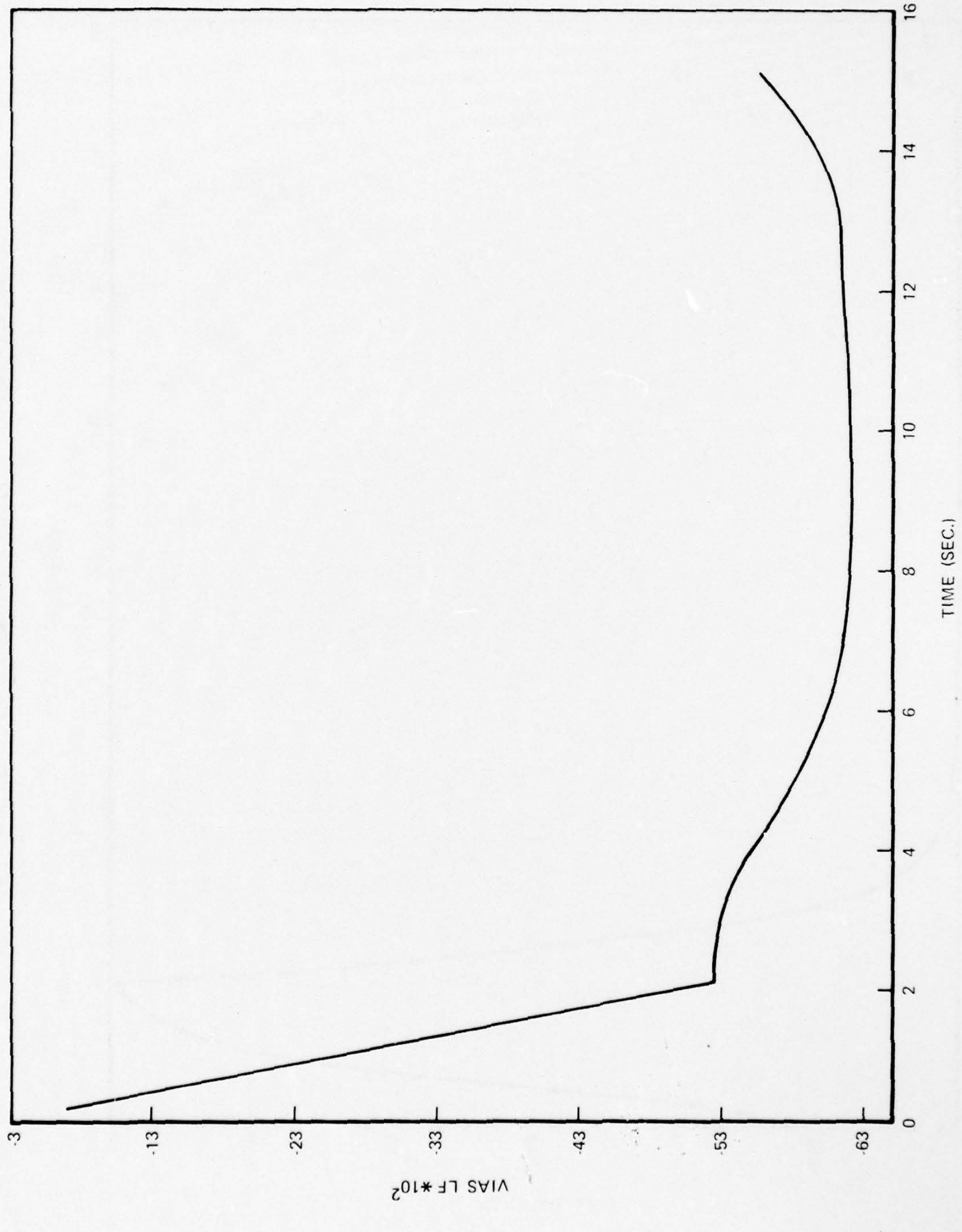


Figure 14. Normal Accelerometer Pickoff Failure

INS vertical accelerometer float leak: This failure was reported as undetectable in Table I from previous investigations. However, for the accelerometer scale factor error $\bar{\xi}^{a}_{33} = \bar{\xi}^{a}_{31} = 0.02$ and above, detection was achieved in the experiments conducted for this effort. Figure 15 portrays a typical result for a scale factor error of 0.04, in which VERT 1, VERT 2, and VERT 3 all clearly surpass their no-failure maximums of -14.5, -5.2 and -2.8, respectively. Times of passage of these thresholds were not always identical, so that checking other likelihood function values when one surpassed its threshold (to determine the probability of their thresholds being surpassed soon) would be required to preclude false alarms.

3.2 SUDDEN FAILURES WITH DRIFTING EFFECTS

As in the previous section, the study of sudden failures with drifting effects will be presented in the same order as Table II so that a direct comparison is readily discernible.

Clogged static line: This failure was detected by both altitude (VERT 1) and vertical velocity (VERT 3) exceeding the no-failure maximum values. As in Table I, the altitude likelihood function indicated the failure before the vertical velocity likelihood function did.

Clogged pitot line: With instrument biases allowed to assume different representative values, this failure could not be discerned consistently, as indicated in Table I. However, if a comparison is made between the Monte Carlo runs with no failures and zeroed instrument biases, and similar runs with no biases but the clogged pitot line simu-

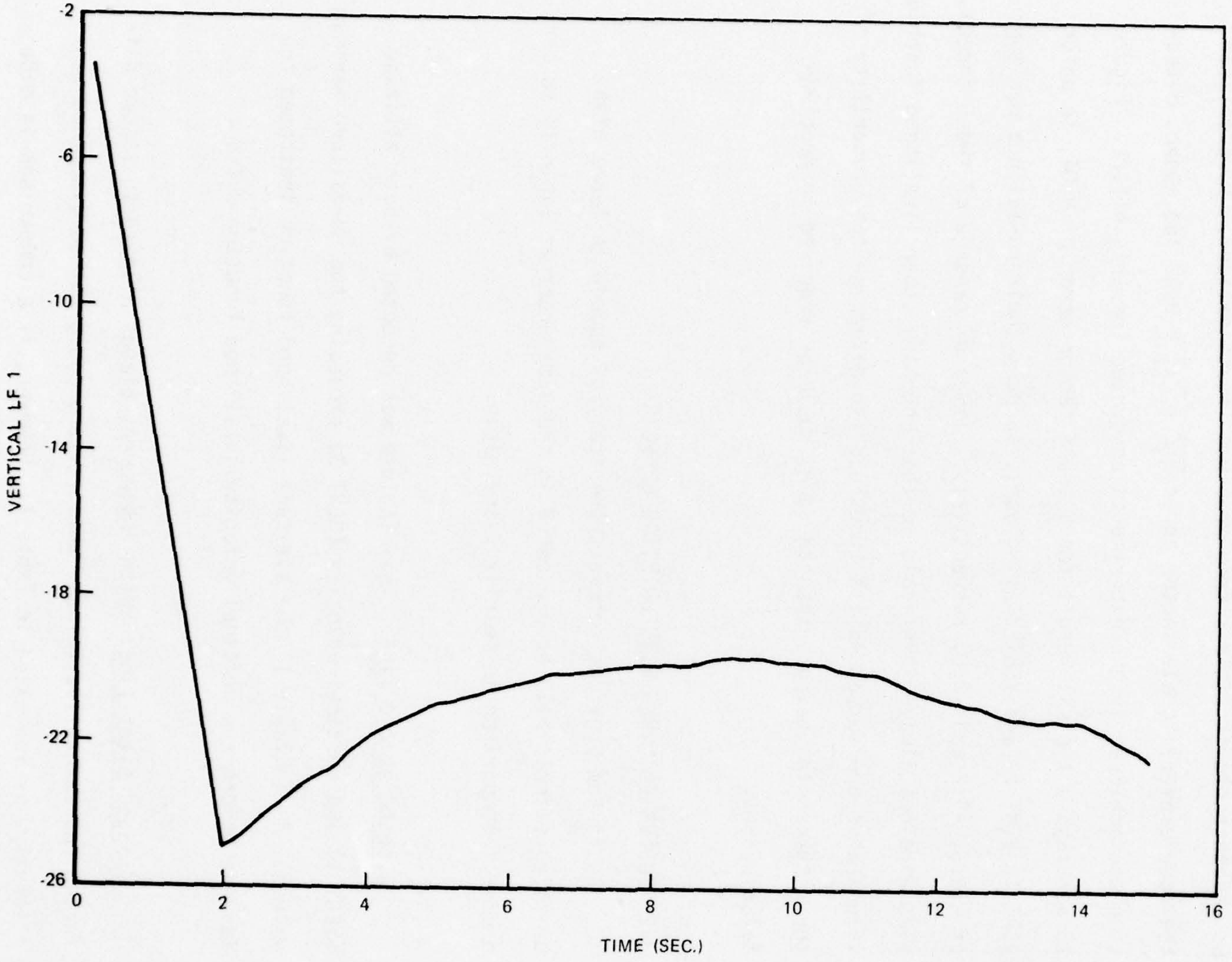


Figure 15a. INS Vertical Accelerometer Float Leak

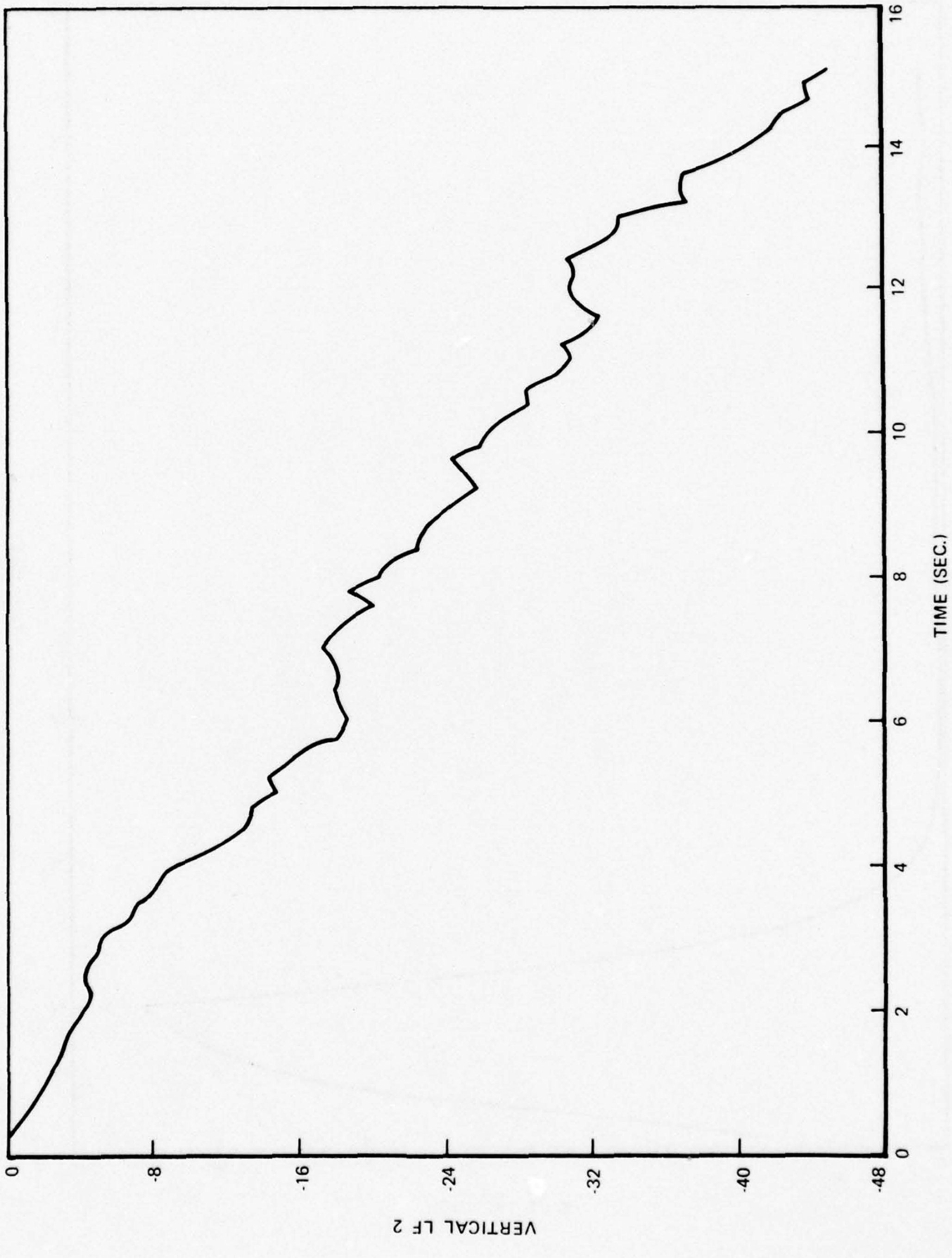


Figure 15b. INS Vertical Accelerometer Float Leak

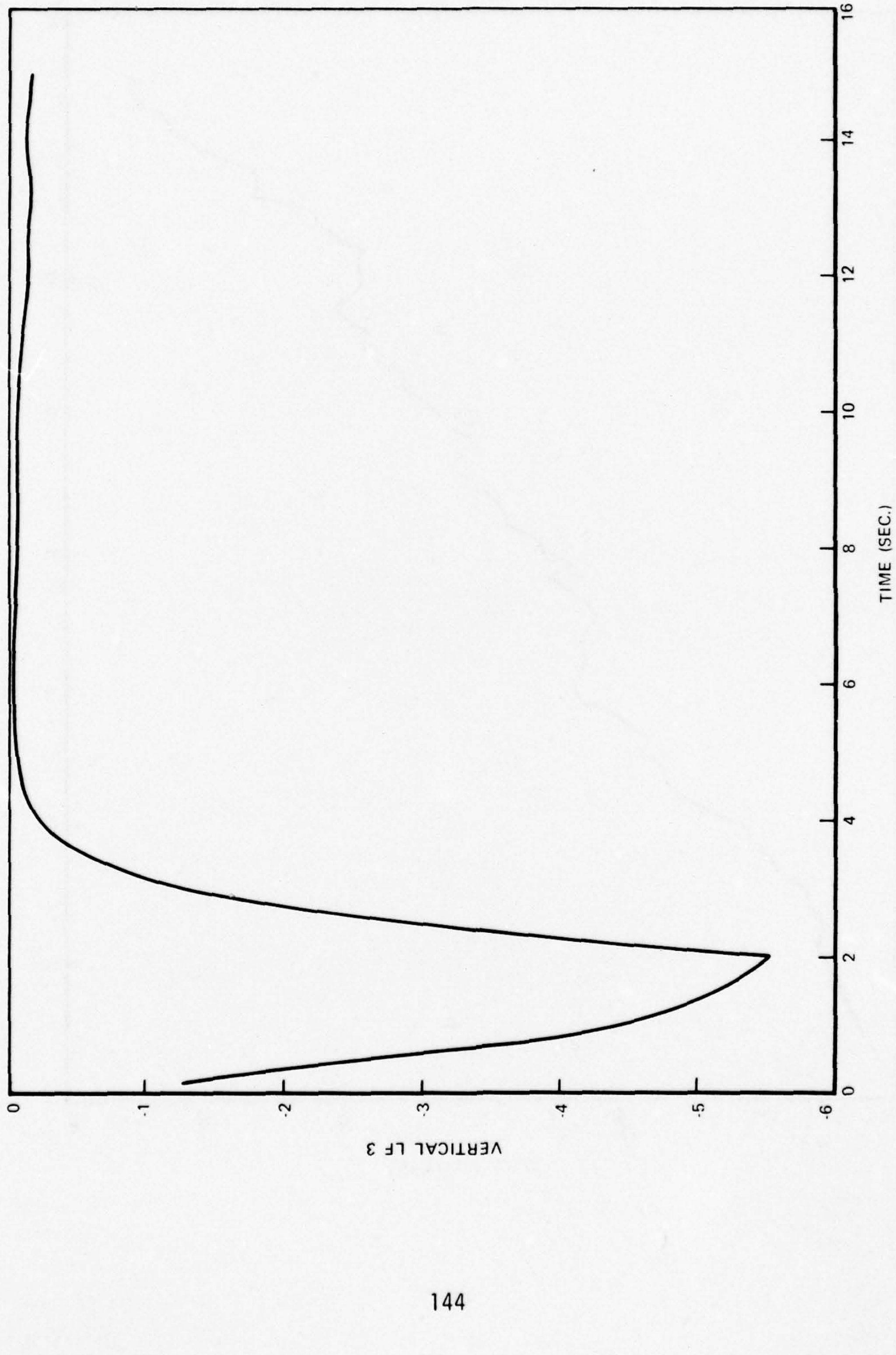


Figure 15c. INS Vertical Accelerometer Float Leak

lated, then the failure is detectable. Figure 8 presented plots of the VIAS likelihood function: in all cases, it converged to magnitudes below -20 in the turn segment and -8 in descent. Note that the original higher values in these plots were due to the initial conditions being established with a trajectory in which instrument biases assumed representative values.

Figure 16 presents the indicated airspeed (VIAS) likelihood function for the case of zero instrument biases and a clogged pitot line. Plot a corresponds to a turn segment, and the likelihood function grows to a value of about -35 (beyond the no-failure maximum of -20). Similarly, plot b corresponds to the descent, and the detectability here is more pronounced: not only is the no-failure maximum value of -8 surpassed, but the growth trend is consistent. In this latter case, the failure would be detected even if the -195 threshold of Table V for descent conditions, or the overall threshold of -430, were used. Nevertheless, its speed of detection would be much improved if instrument biases were compensated in preflight as suggested in Section III (allowing a tighter threshold to be used). In fact, such compensation would be requisite for the detection of a clogged pitot line during the turn segment in most of the Monte Carlo runs, though not for detection in descent.

INS vertical gyro torquer failure: This failure in level flight would be expected to affect the INS pitch and roll (INS 1 and INS 2) likelihood functions. Figure 17 presents representative plots of these

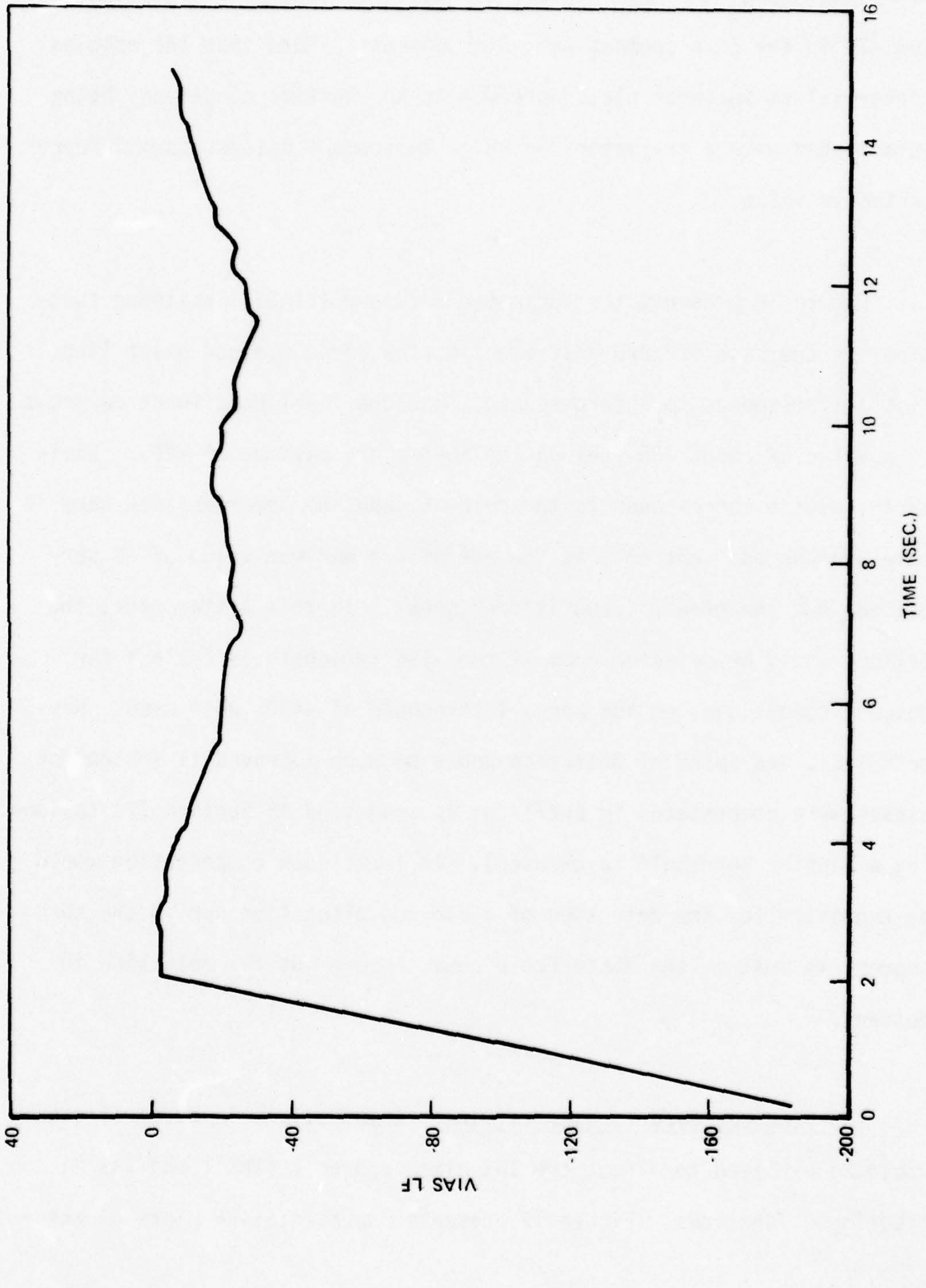


Figure 16a. Clogged Pitot Line in Turn

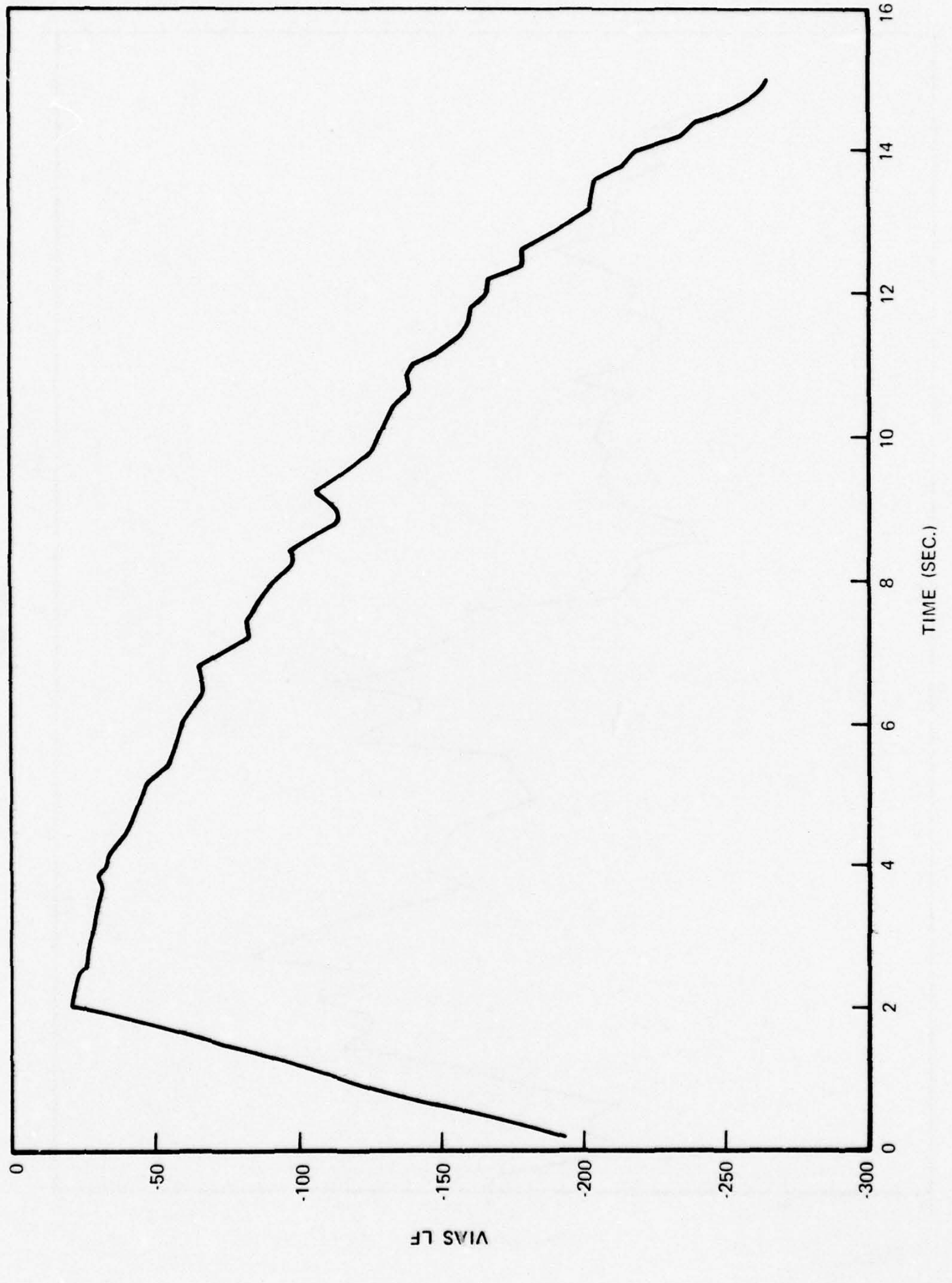


Figure 16b. Clogged Pitot Line in Descent

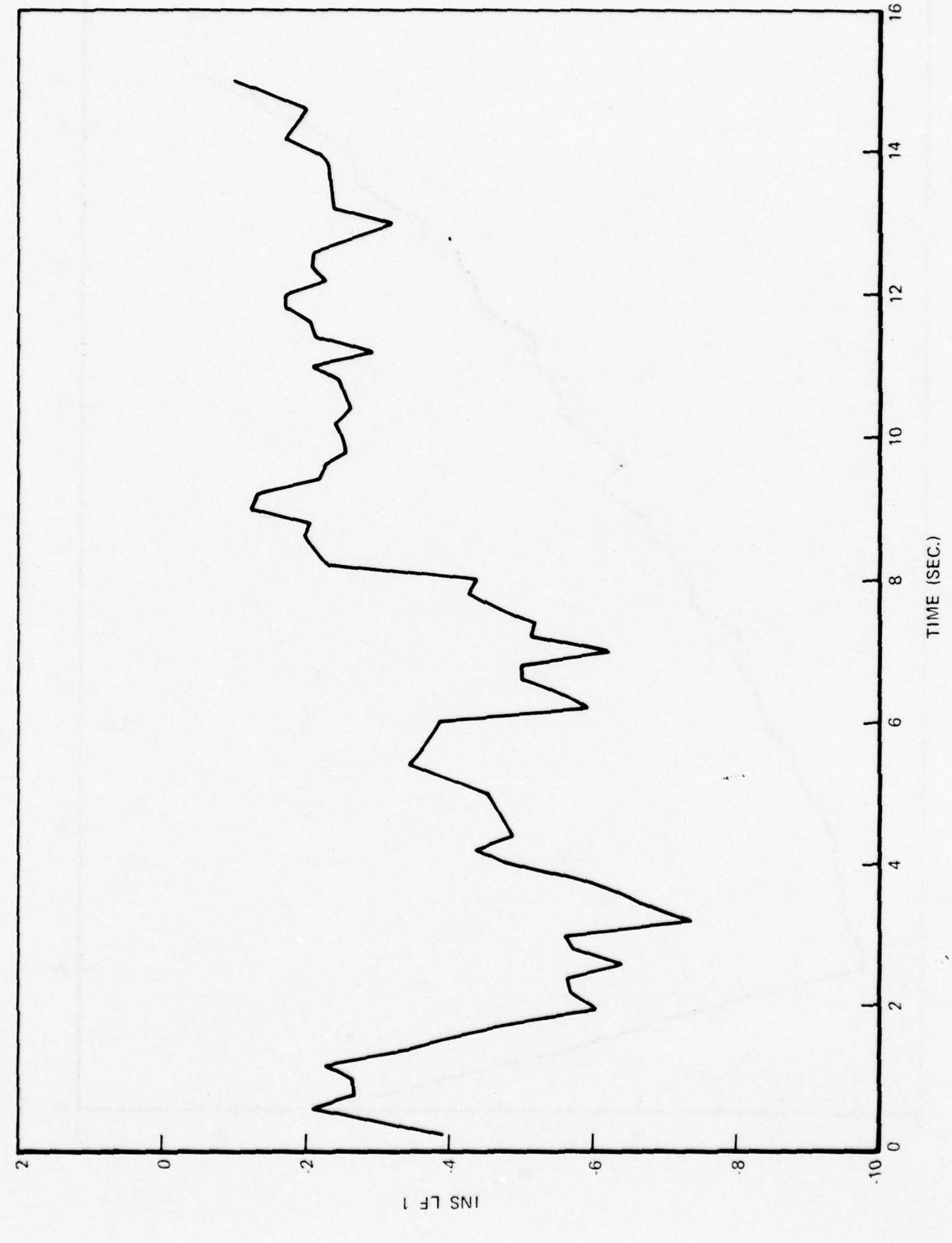


Figure 17a. INS Vertical Gyro Torquer Failure

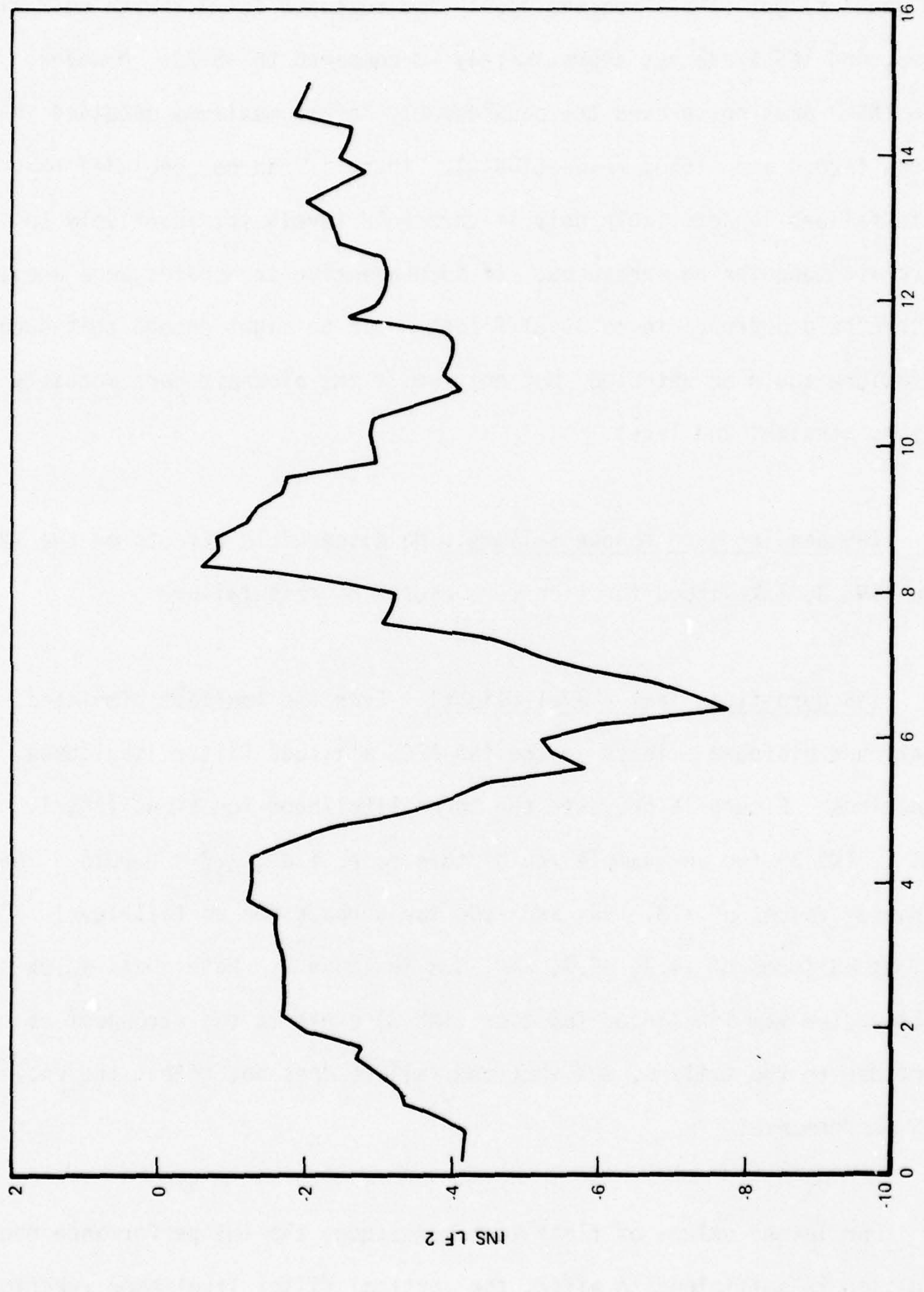


Figure 17b. INS Vertical Gyro Torquer Failure

two functions. These two graphs do exceed the no-fail maximums achieved in level flight (INS 1 reaches about -7.5 compared to -4.2 with no failures, and INS 2 reaches approximately -8 compared to -5.7). However, the INS 2 does not exceed the considerably larger maximums obtained in turns (-20.4 and -1650, respectively). Thus, it can be concluded that this failure is detectable only if threshold levels set adaptively to aircraft maneuvering were used. If such adaptive thresholds were used, a threshold appropriate to level flight would be tight enough that such a failure could be detected, but only while the aircraft were actually flying straight and level.

INS heading gyro torque failure: No discernible effects on the INS yaw (INS 3) likelihood function were caused by this failure.

INS gyro float leak (level flight): Even the smallest simulated leaks had profound effects on the INS-AFCS attitude filter likelihood functions. Figure 18 presents the three likelihood functions (INS 1, INS 2, INS 3) for an example run of this case, i.e., $\overline{\xi}_{11}^{g}$ = 0.0025. The achieved values of -13, -90, and -800 far surpass the no-fail level flight maximums of -4.2, -5.7, and -5.8 in Table V. Note that, as expected, the yaw likelihood function (INS 3) exhibits the strongest effect due to the failure, but that the failure does not affect the entire INS performance.

For larger values of float leak magnitude, the INS performance degradation is sufficient to affect the vertical filter likelihood functions through the influence of the INS vertical accelerometer. For instance,

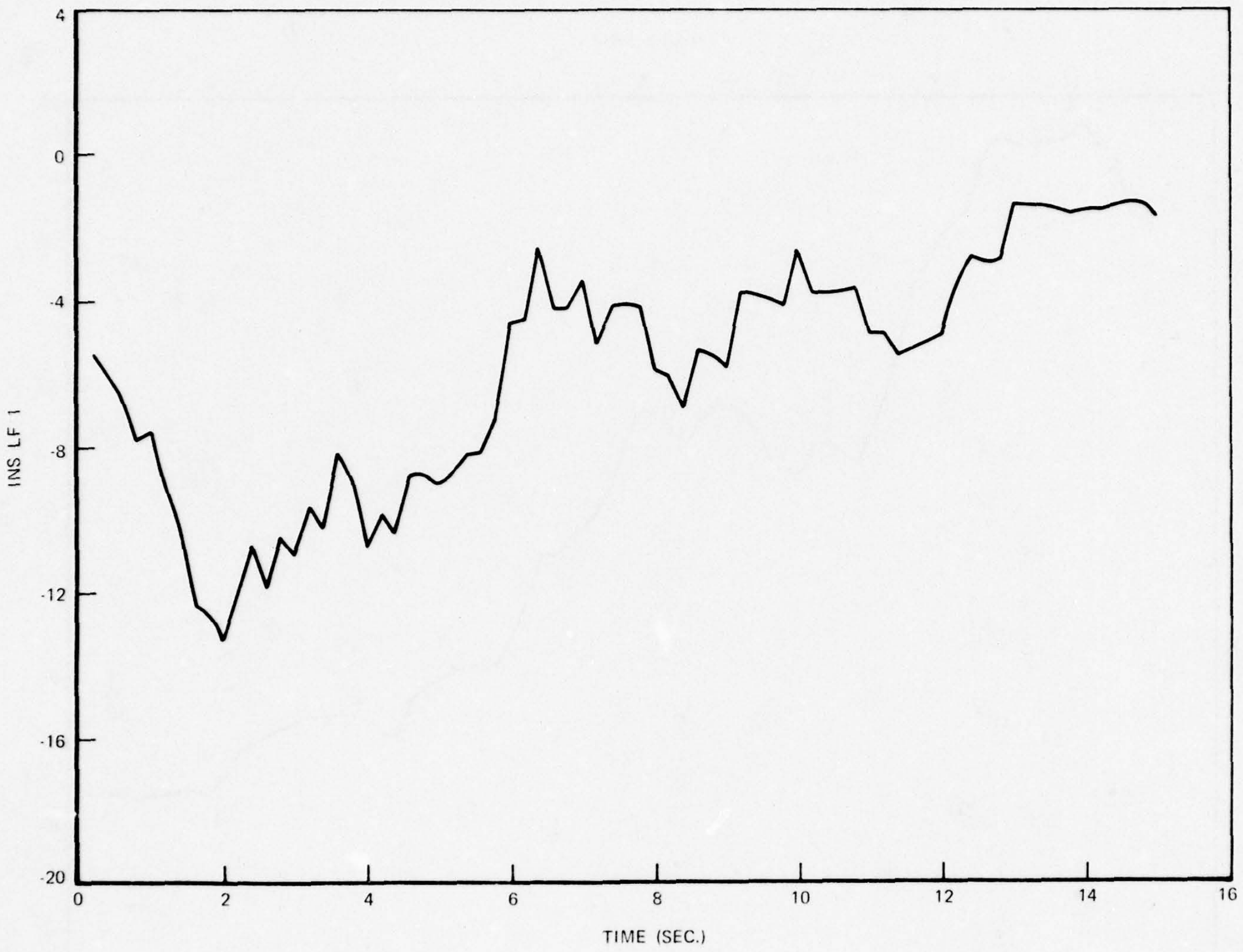


Figure 18a. INS Gyro Float Leak in Level Flight

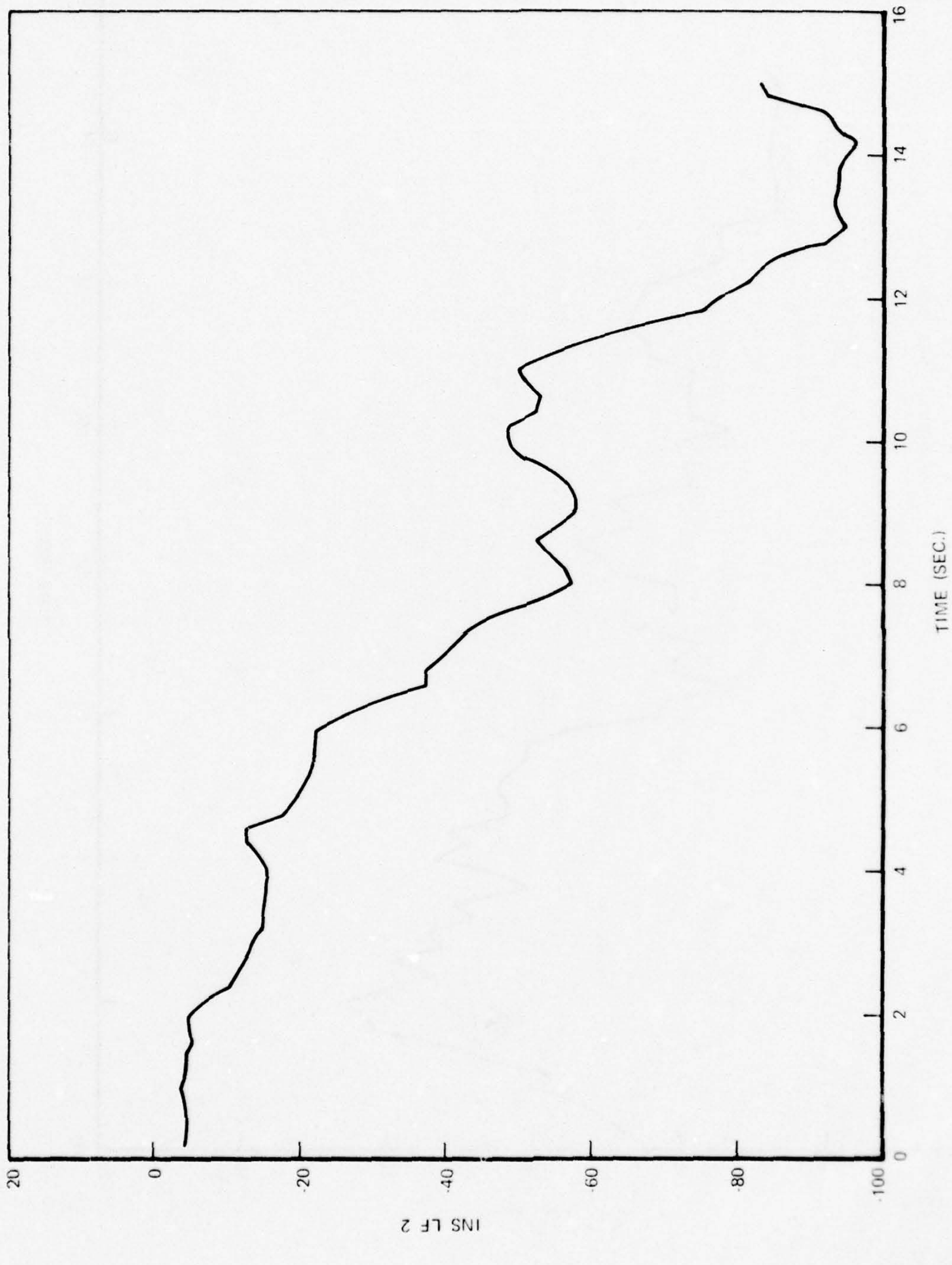


Figure 18b. INS Gyro Float Leak in Level Flight

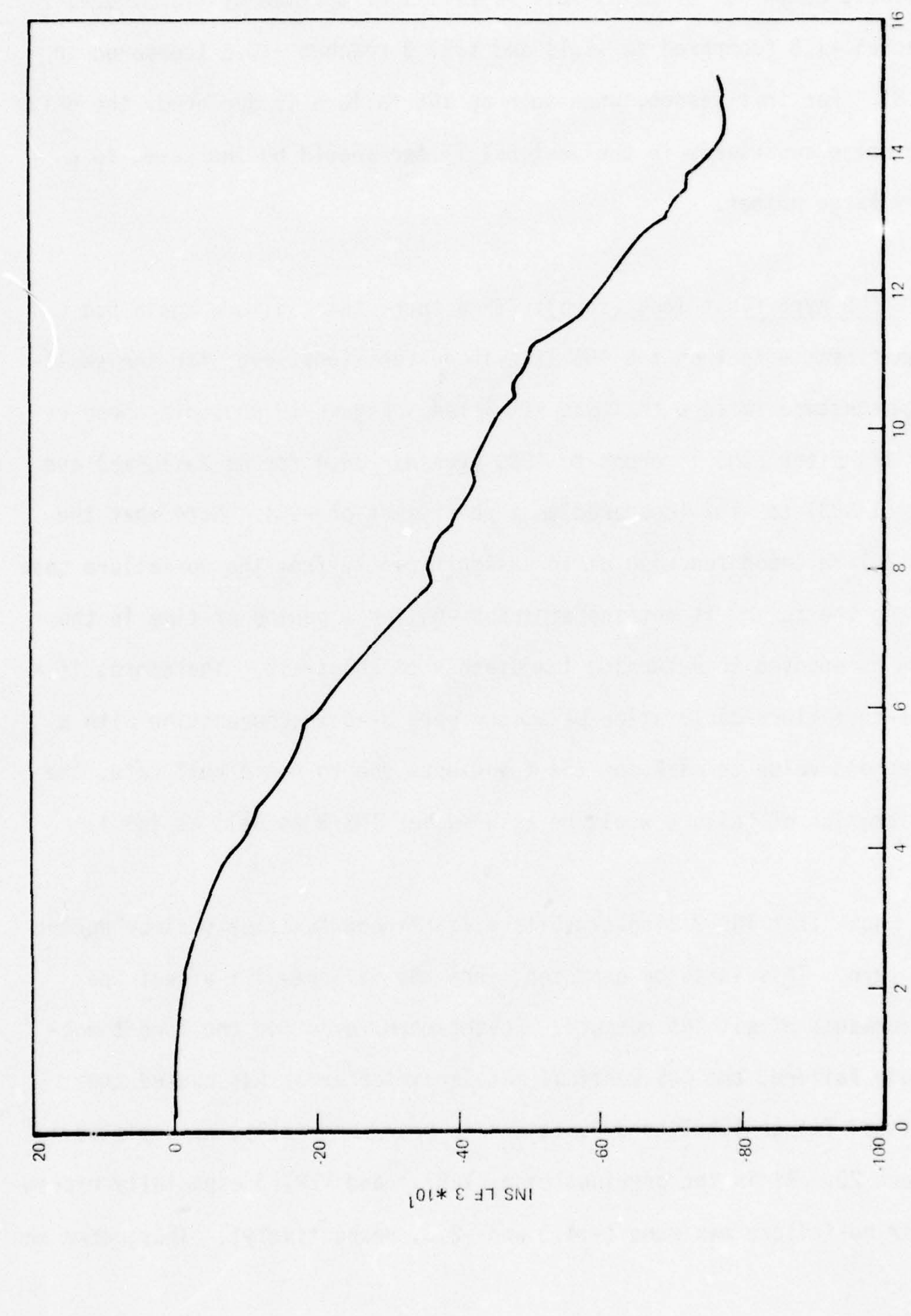


Figure 18c. IMS Gyro Float Leak in Level Flight

when $\bar{\xi}^{g}_{11}$ = 0.1, VERT 1 reaches -42 in two seconds from start of the failure, compared to its no-fail level flight maximum of -13.3, VERT 2 reaches -1.5 (compared to -1.1) and VERT 3 reaches -10.6 (compared to -2.8). For this reason, when such an INS failure is declared, the driving noise covariance in the vertical filter should be increased to a very large number.

INS gyro float leak (turn): In a turn, this failure again had significant effect on the INS likelihood functions, even for the smallest magnitude failure that was simulated. Figure 19 presents these results: pitch (INS 1) drops to -230 (versus -20.4 for no failures) and yaw (INS 3) to -102 (compared to a peak value of -96). Note that the INS 3 likelihood function differs significantly from the no-failure case during the turn: it remains at about -60 for a period of time in the turn as opposed to returning immediately to about -10. Therefore, if a time-to-failure-declaration parameter were used in conjunction with a threshold value to mask out the transients due to rapid roll rate, the declaration of failure would be assured for INS 3 as well as INS 1.

Note that INS 2 also exhibits a likelihood function buildup during the turn. This is to be expected since the failure will affect the performance of all INS outputs. Furthermore, even for the lowest magnitude failure, the INS vertical accelerometer error has caused the vertical filter likelihood functions to grow abnormally, as depicted in Figure 20. As in the previous case, VERT 1 and VERT 3 especially exceed their no-failure maximums (-14.0 and -2.5, respectively). Thus, when an

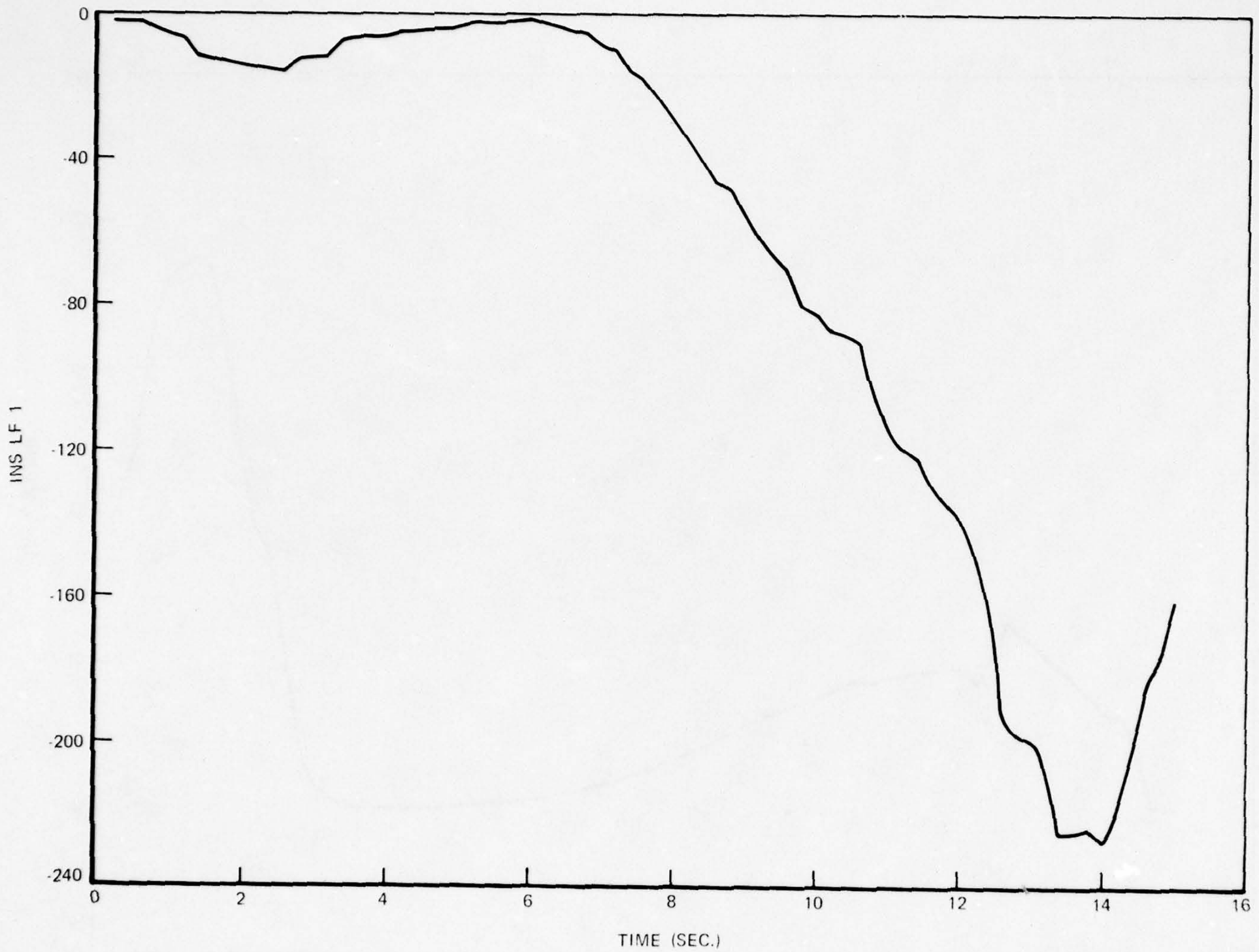


Figure 19a. INS Gyro Float Leak in Turn-Attitudes

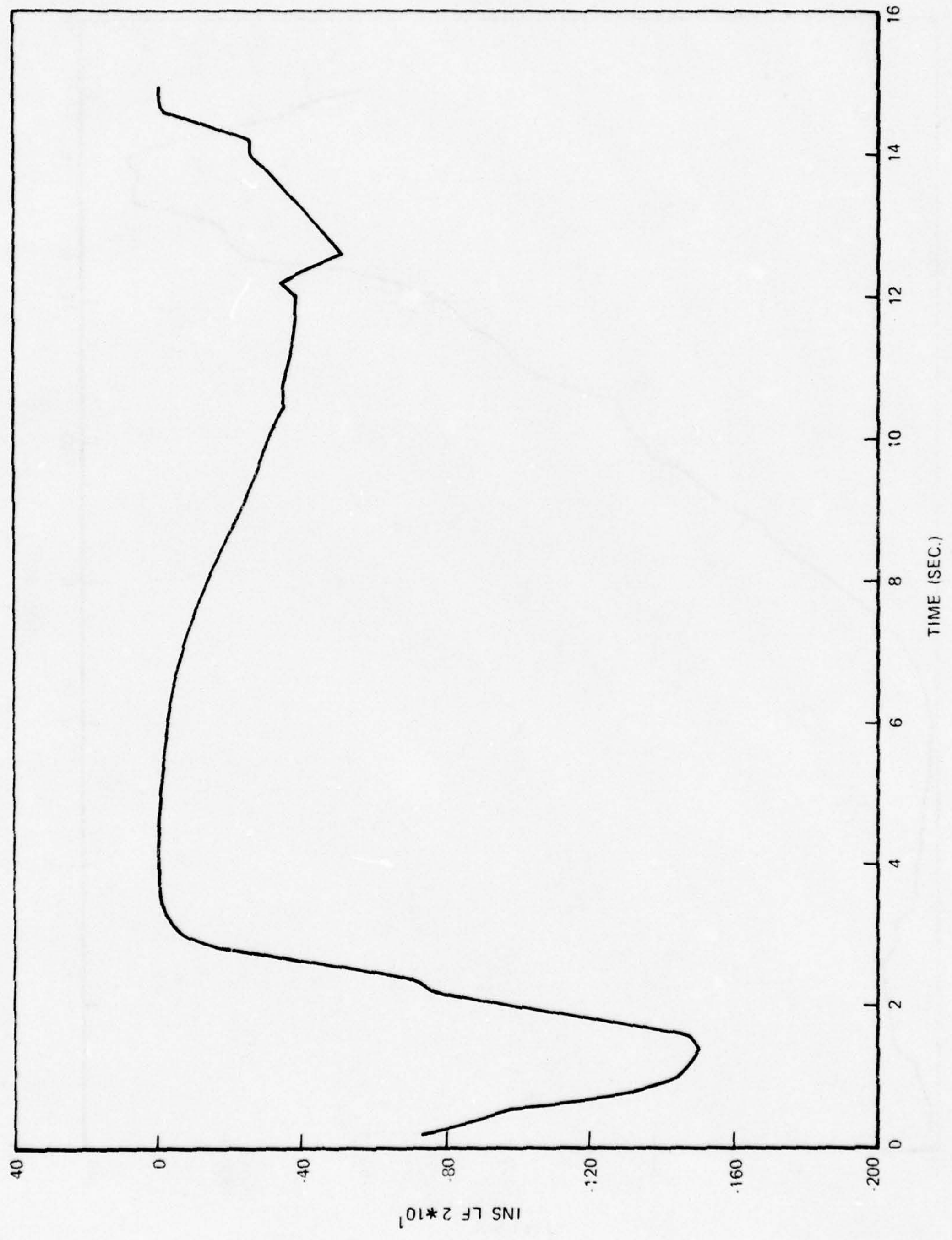


Figure 19b. INS Gyro Float Leak in Turn-Attitudes

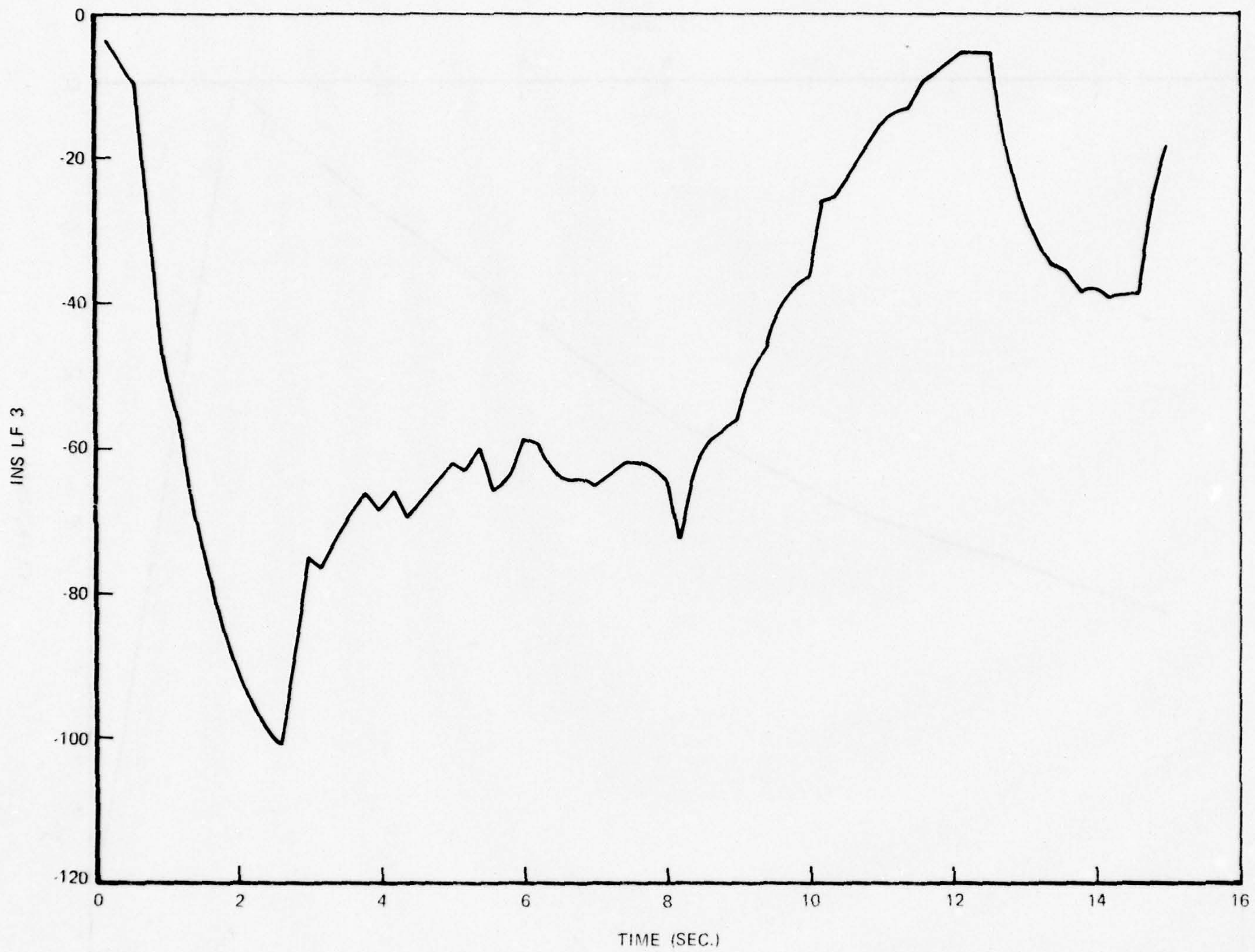


Figure 19c. INS Gyro Float Leak in Turn-Attitudes

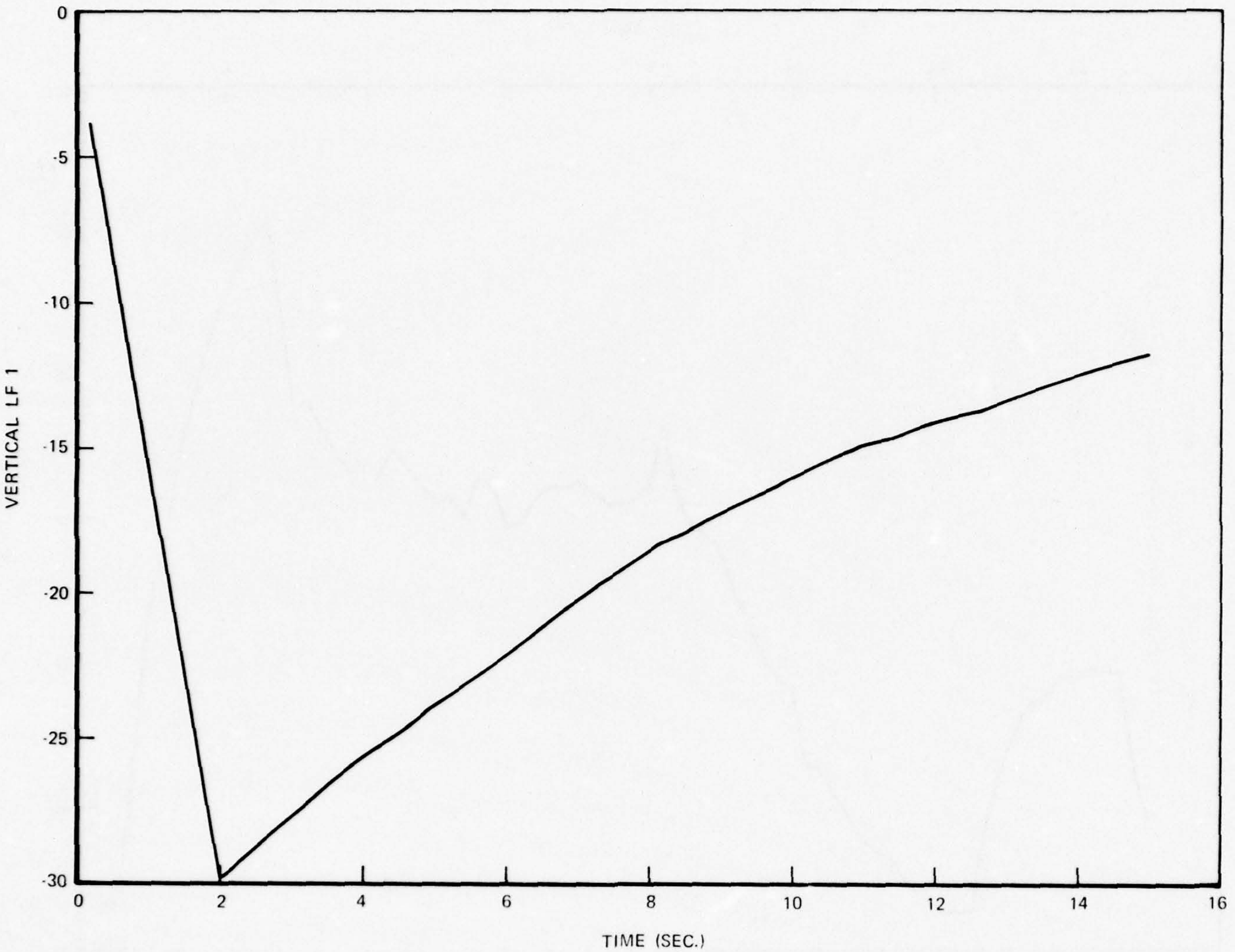


Figure 20a. INS Gyro Float Leak in Turn

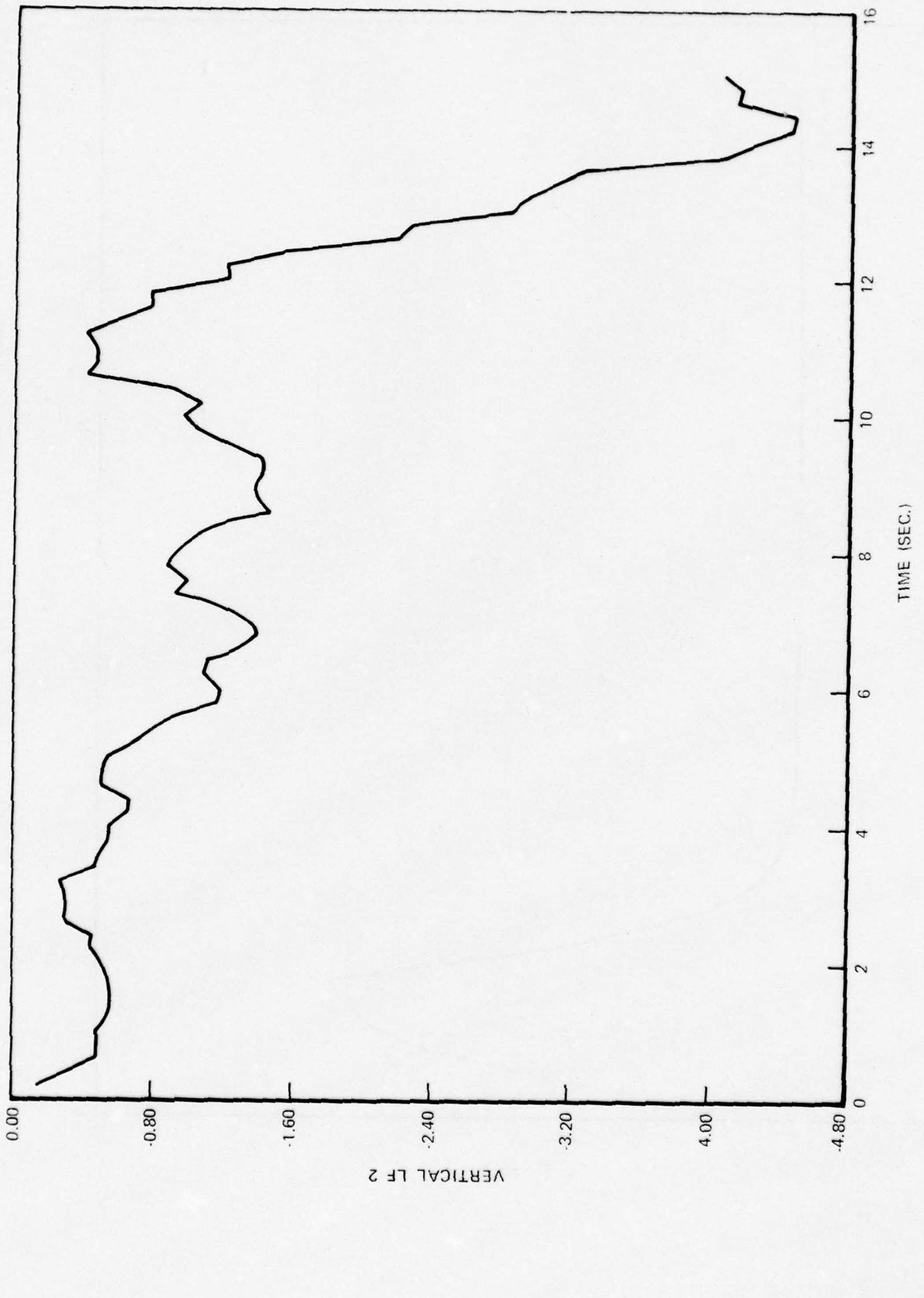


Figure 20b. INS Gyro Float Leak in Turn

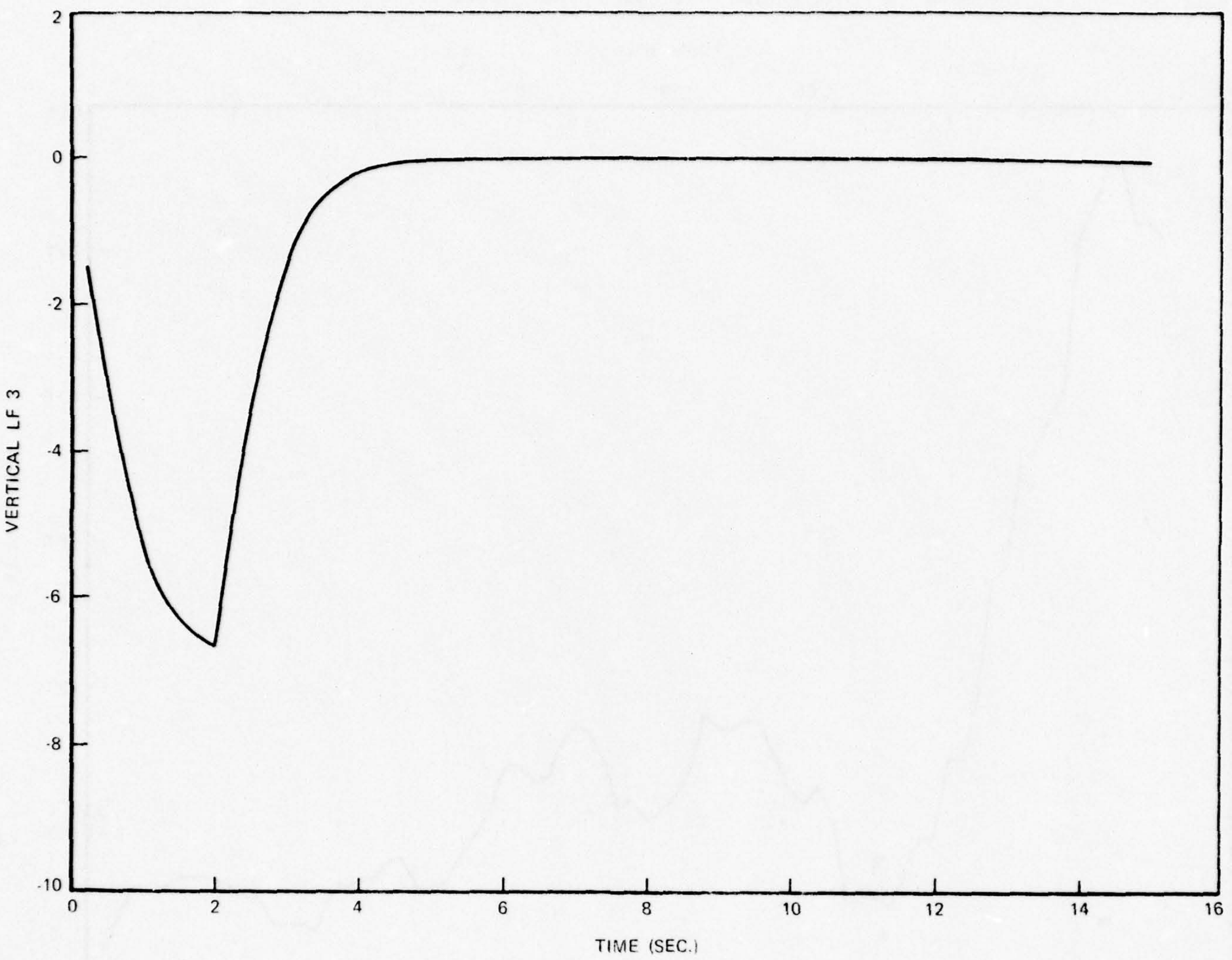


Figure 20c. INS Gyro Float Leak in Turn

INS failure is sensed, either the driving noise variance for the vertical filter must be increased or the failure detection based on the vertical filter likelihood functions must be disenabled.

Increasing the magnitude of the failure makes the effects depicted above more pronounced. Thus, the rapid detection in Table II is possible. (For instance, when $\bar{\xi}_{11}^{g} = 0.005$, INS 3 (yaw) reaches -5000 in three seconds!). The "false alarms" reported in that table are due to the propagation of the degraded INS performance into the VERT 2 likelihood function and the entire vertical filter.

<u>Loss of cutoff for the vertical gyro</u>: During a turn, such a failure should be declared by the AHRS roll (AHRS 2) likelihood function surpassing threshold. With loss of good vertical gyro performance, it would be expected that AHRS pitch (AHRS 1) would also grow. Figure 21 presents a typical set of plots of these two likelihood functions. If a time-to-failure-declaration parameter or some other method were used with a threshold value to mask out normal transients in the AHRS 2 likelihood function, then the growth during the turn itself to a value of about -350 would be detectable as a failure. As expected, the AHRS 1 likelihood has surpassed its no-failure maximum value of -36.

<u>Loss of cutoff for directional gyro</u>: No effects could be observed in the AHRS yaw (AHRS 3) likelihood function when this failure was simulated.

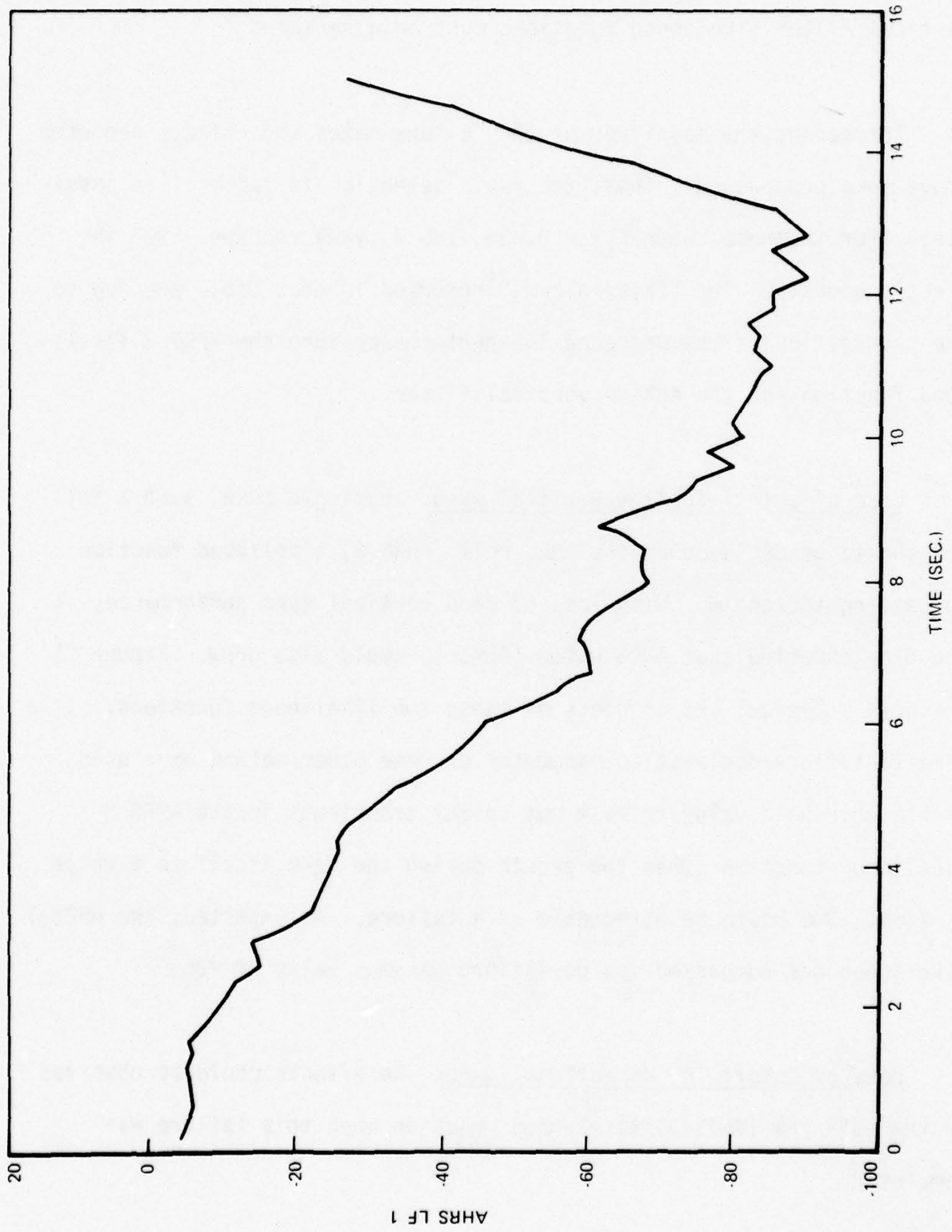


Figure 21a. Loss of Cutoff for Vertical Gyro

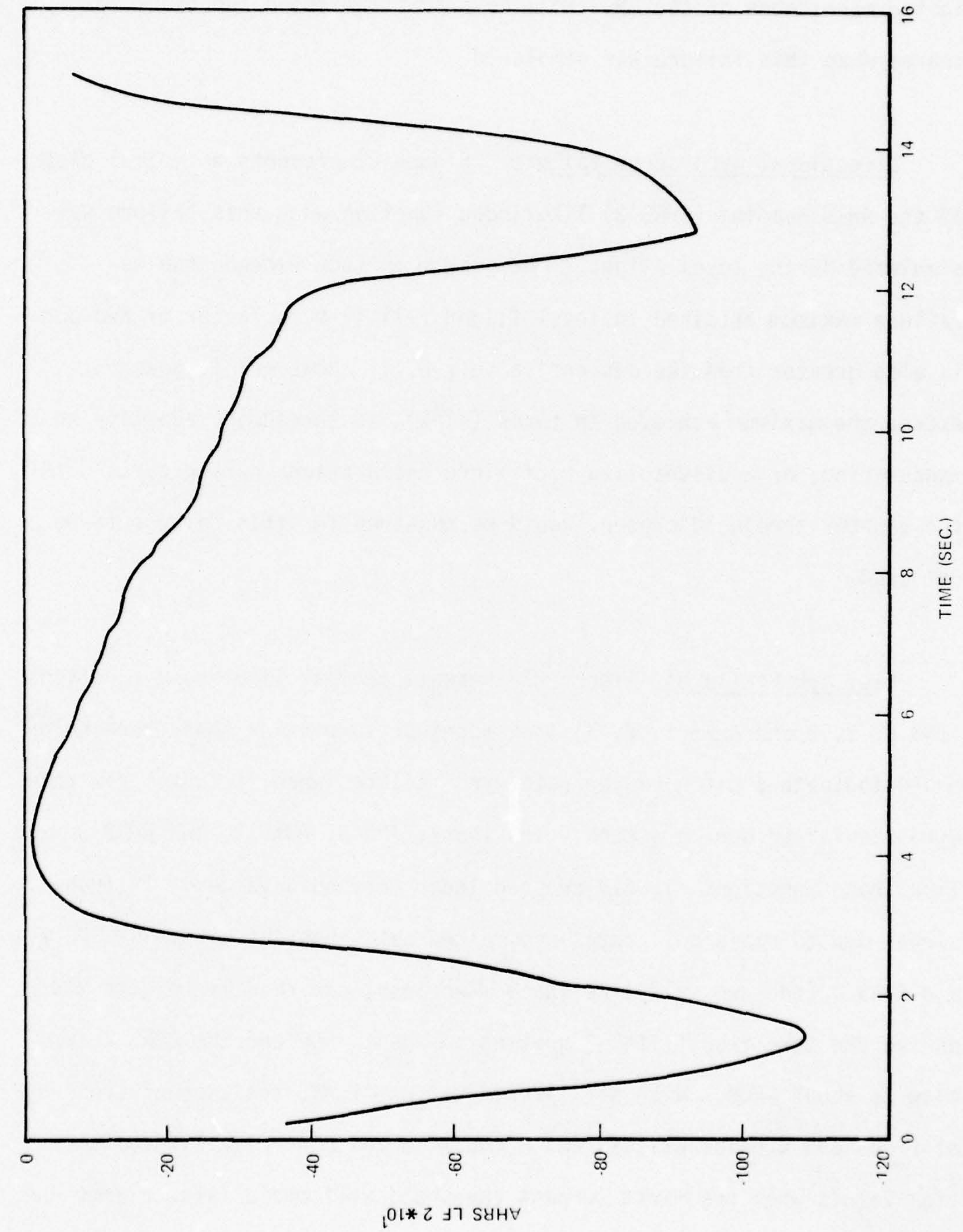


Figure 21b. Loss of Cutoff for Vertical Gyro

Vertical gyro servo failure: No consistently discernible increase in the magnitudes of the AHRS roll or pitch likelihood functions appeared when this failure was simulated.

Directional gyro servo failure: Figure 22 presents a typical plot of the AHRS heading (AHRS 3) likelihood function when this failure was simulated during level flight. The peak magnitude exceeds the no-failure maximum attained in level flight (-11.1) by a factor of two and is also greater than the descent value (-6.1). However, it does not exceed the maximum achieved in turns (-145), so thresholds adaptive to maneuvering, or a disenabling of failure declarations during turns with the tighter threshold chosen, would be required for this failure to be detected.

Rate gyro failure: Figure 23 presents the six likelihood functions (INS 1, 2, 3 and AHRS 1, 2, 3) that together surpassing their thresholds would indicate a pitch or yaw rate gyro failure, when in fact a yaw rate gyro was failed during a turn. The INS 1, INS 3, AHRS 1, and AHRS 3 likelihood functions clearly crossed their threshold values. If transients due to rapid roll rates are masked out, then the growth of INS 2 and AHRS 2 (the two roll likelihood functions) can readily be detected during the turn itself, INS 2 growing to about -500 and the AHRS 2 function to about -200. With six likelihood functions, the time of crossing of threshold problem exists, and a check on the other likelihood function values when the first surpass threshold will avoid false alarms due to this problem.

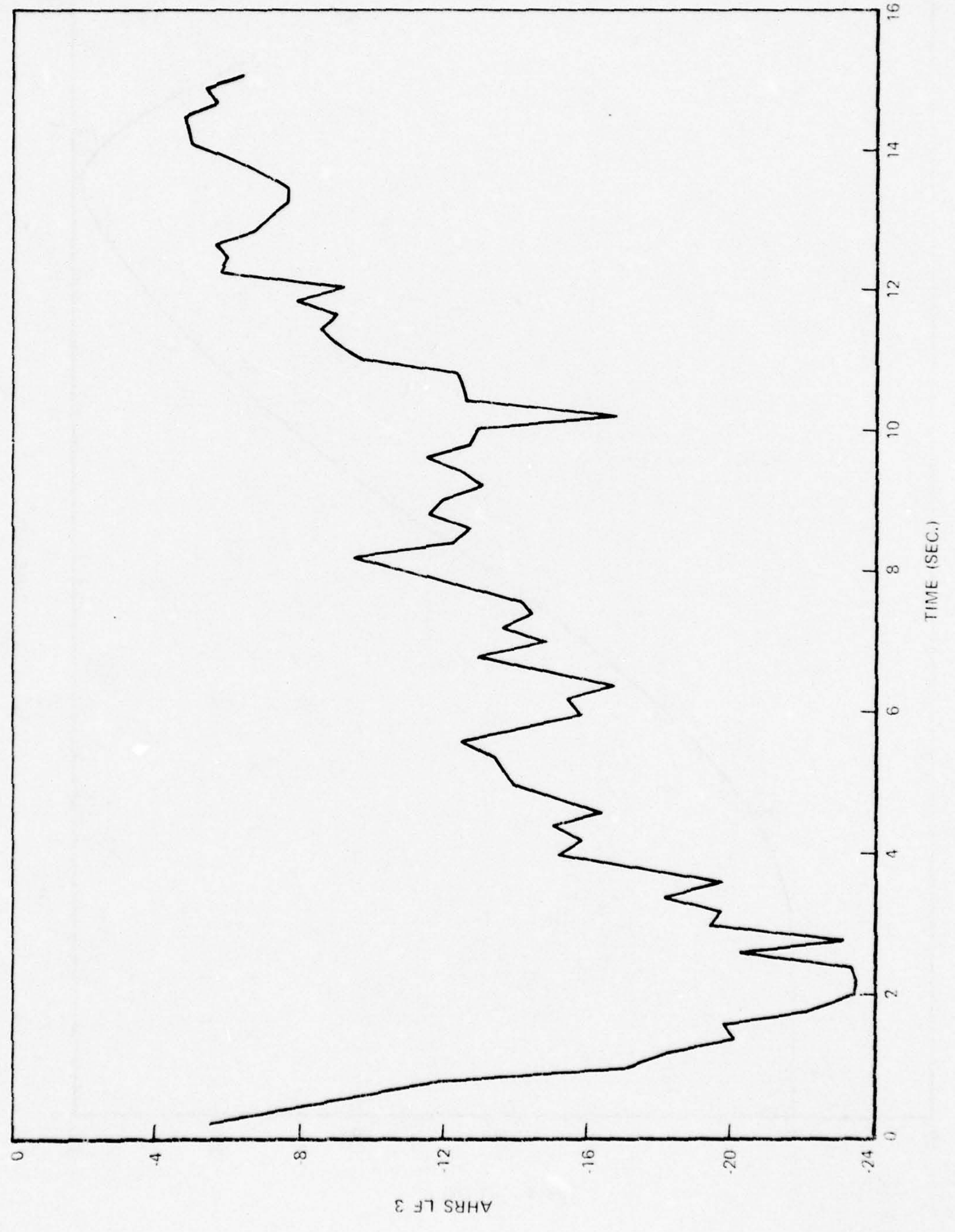


Figure 22. Directional Gyro Servo Failure

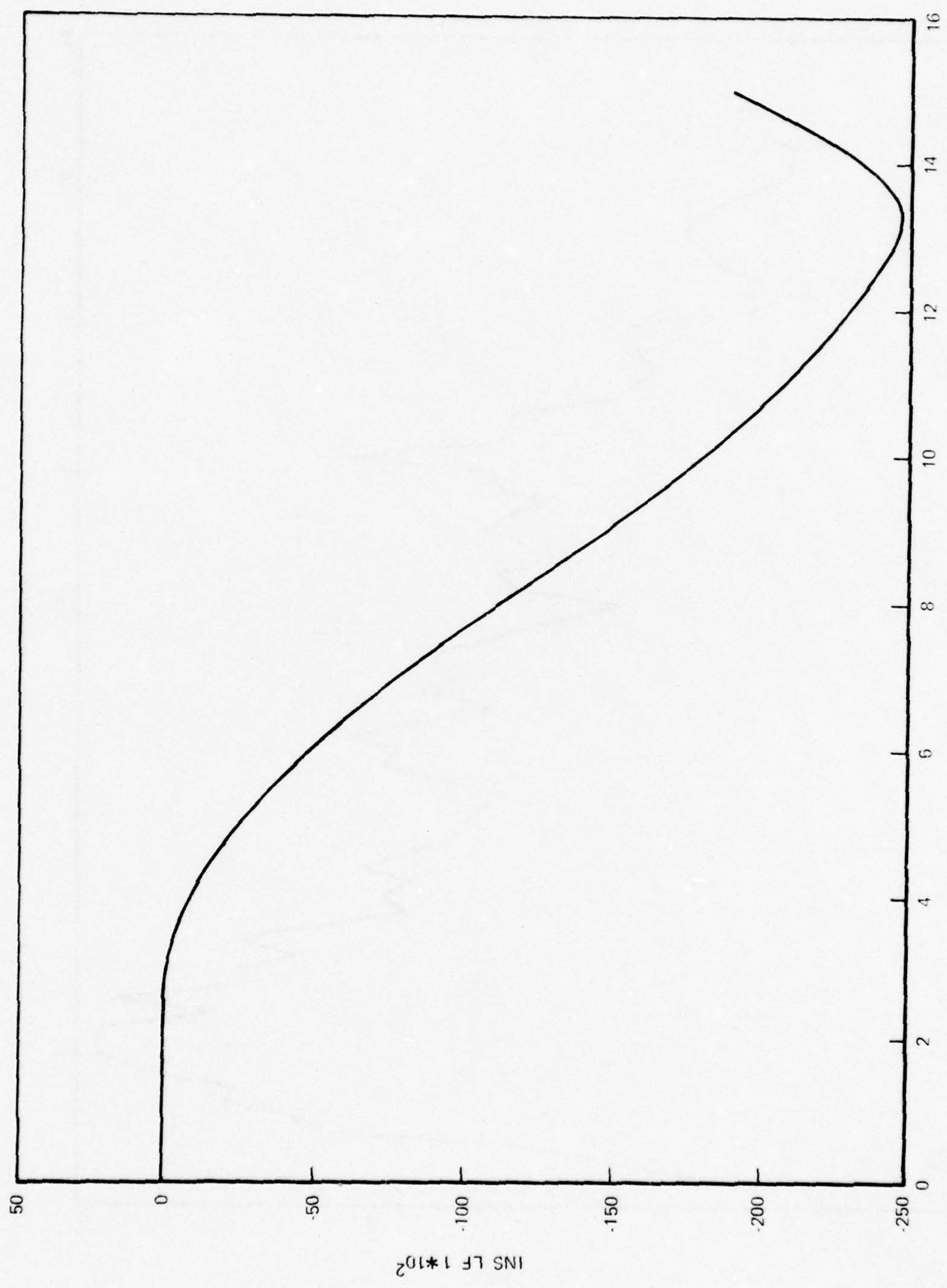


Figure 23a. Yaw Rate Gyro Failure

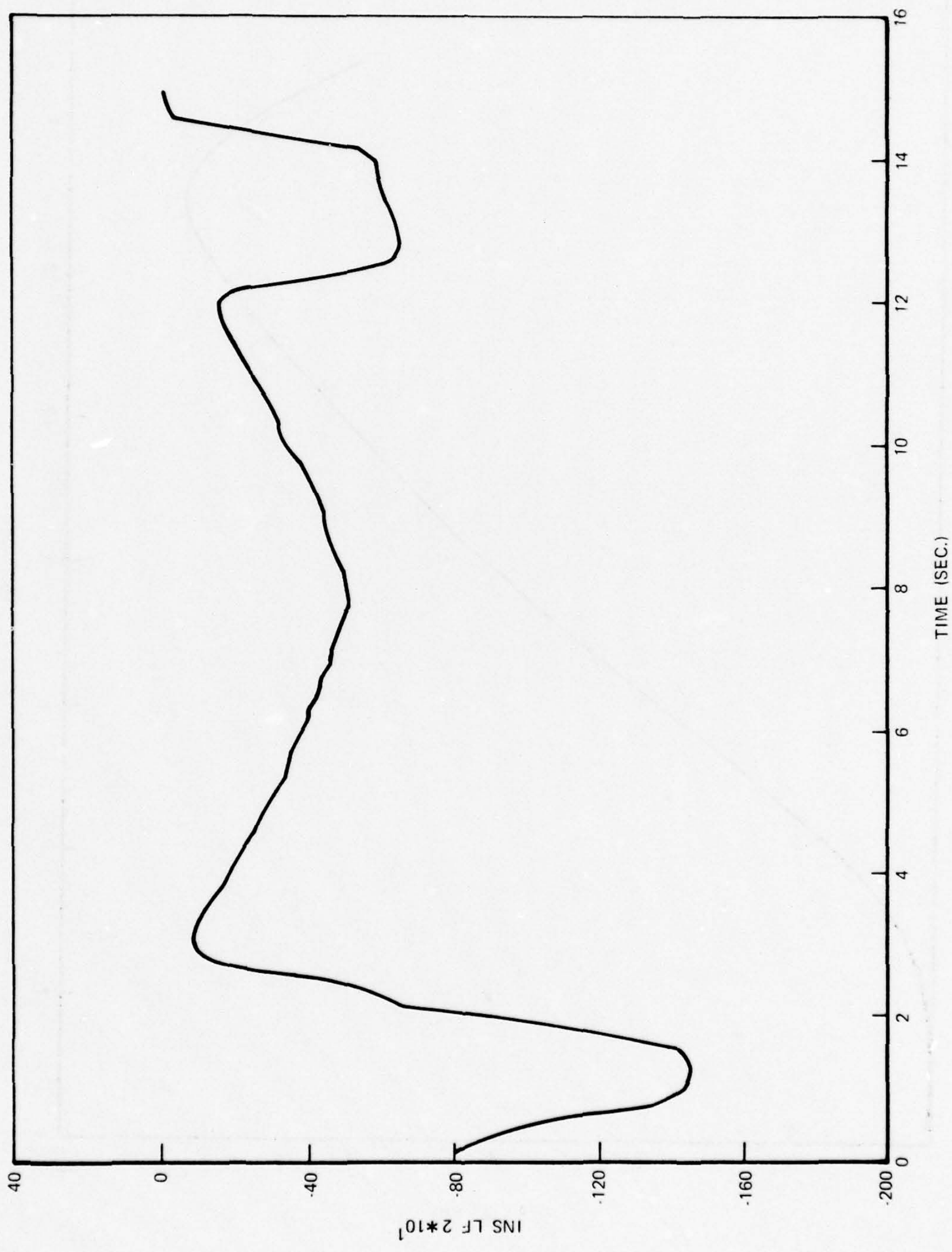


Figure 23b. Yaw Rate Gyro Failure

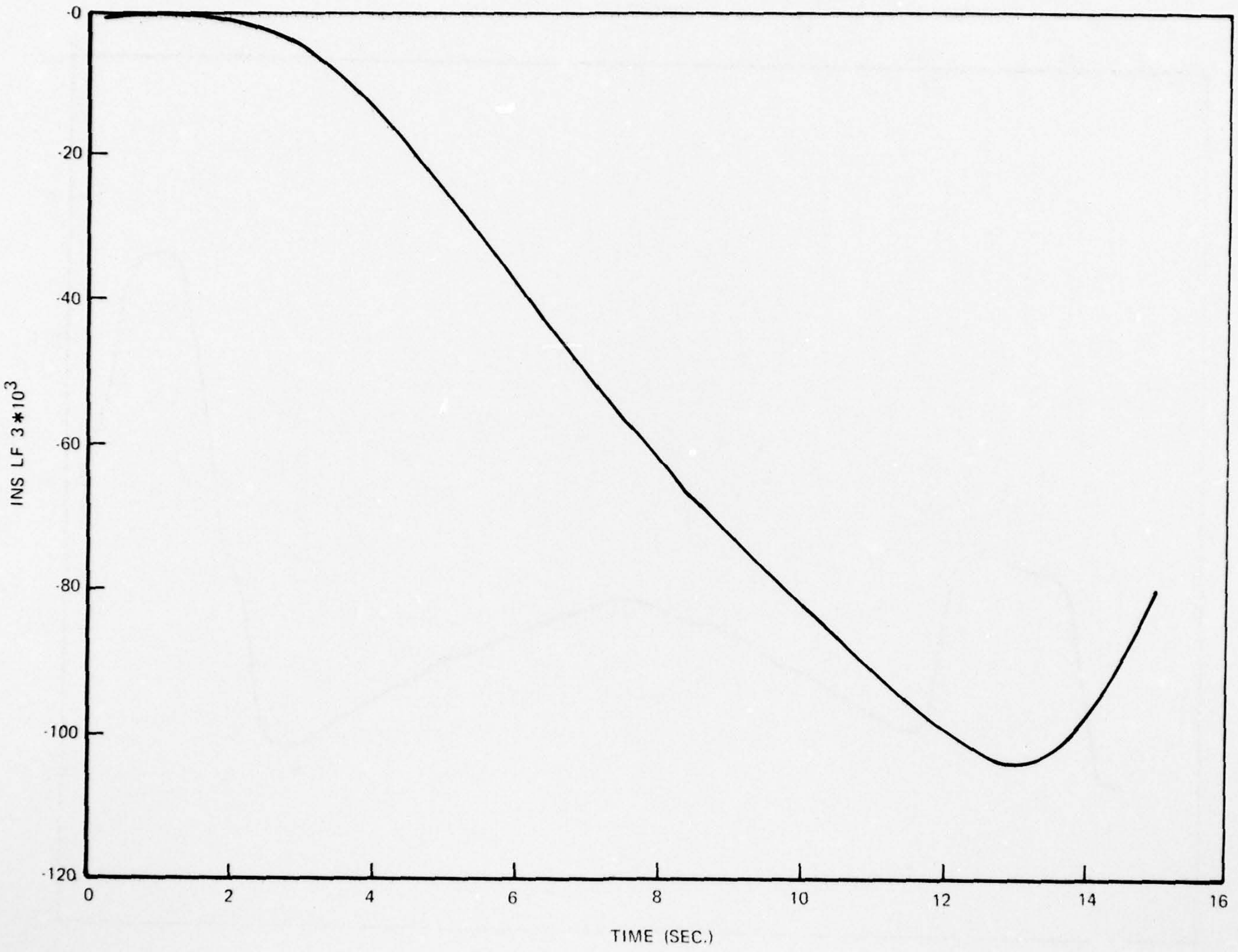


Figure 23c. Yaw Rate Gyro Failure

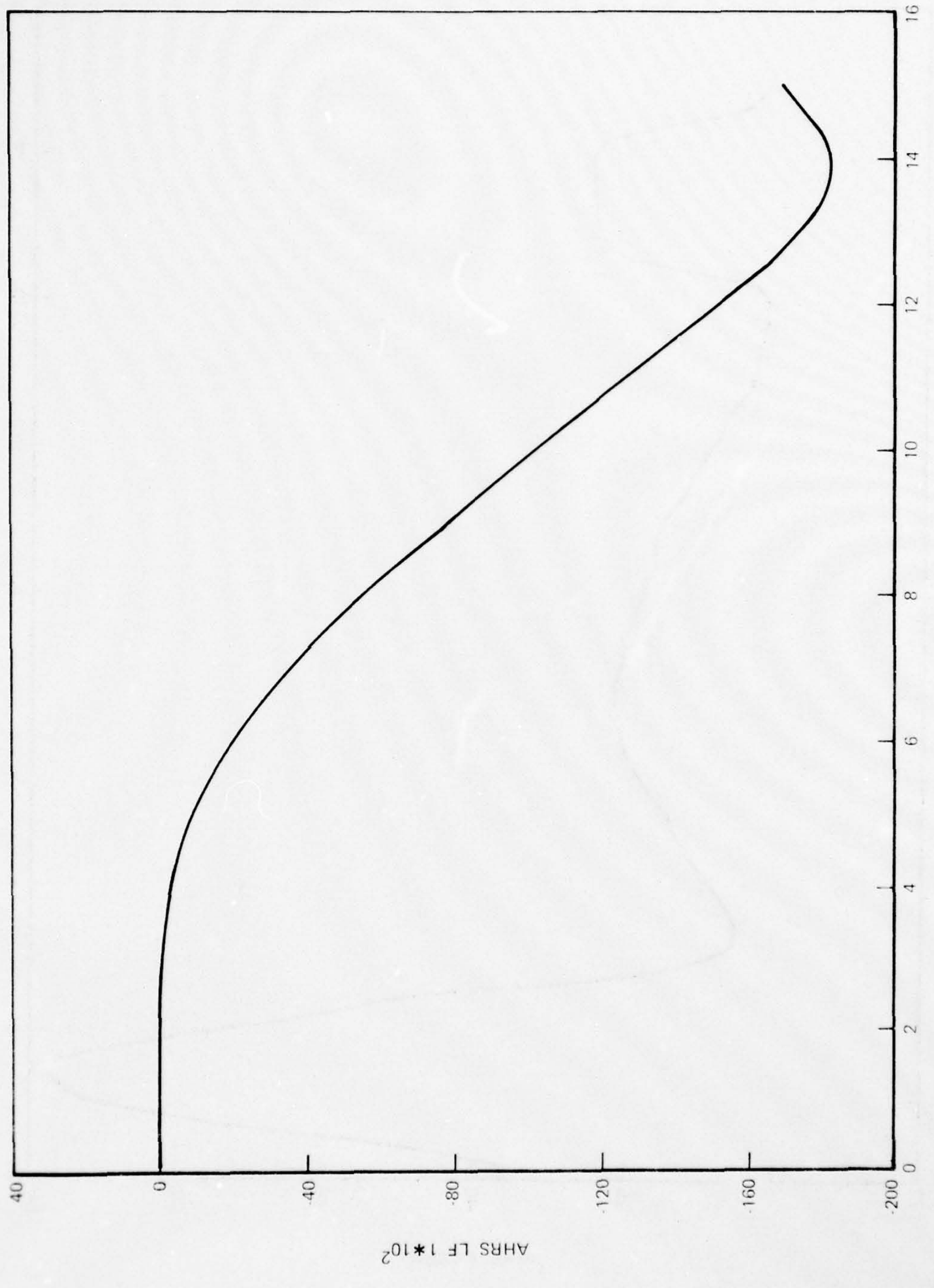


Figure 23d. Yaw Rate Gyro Failure

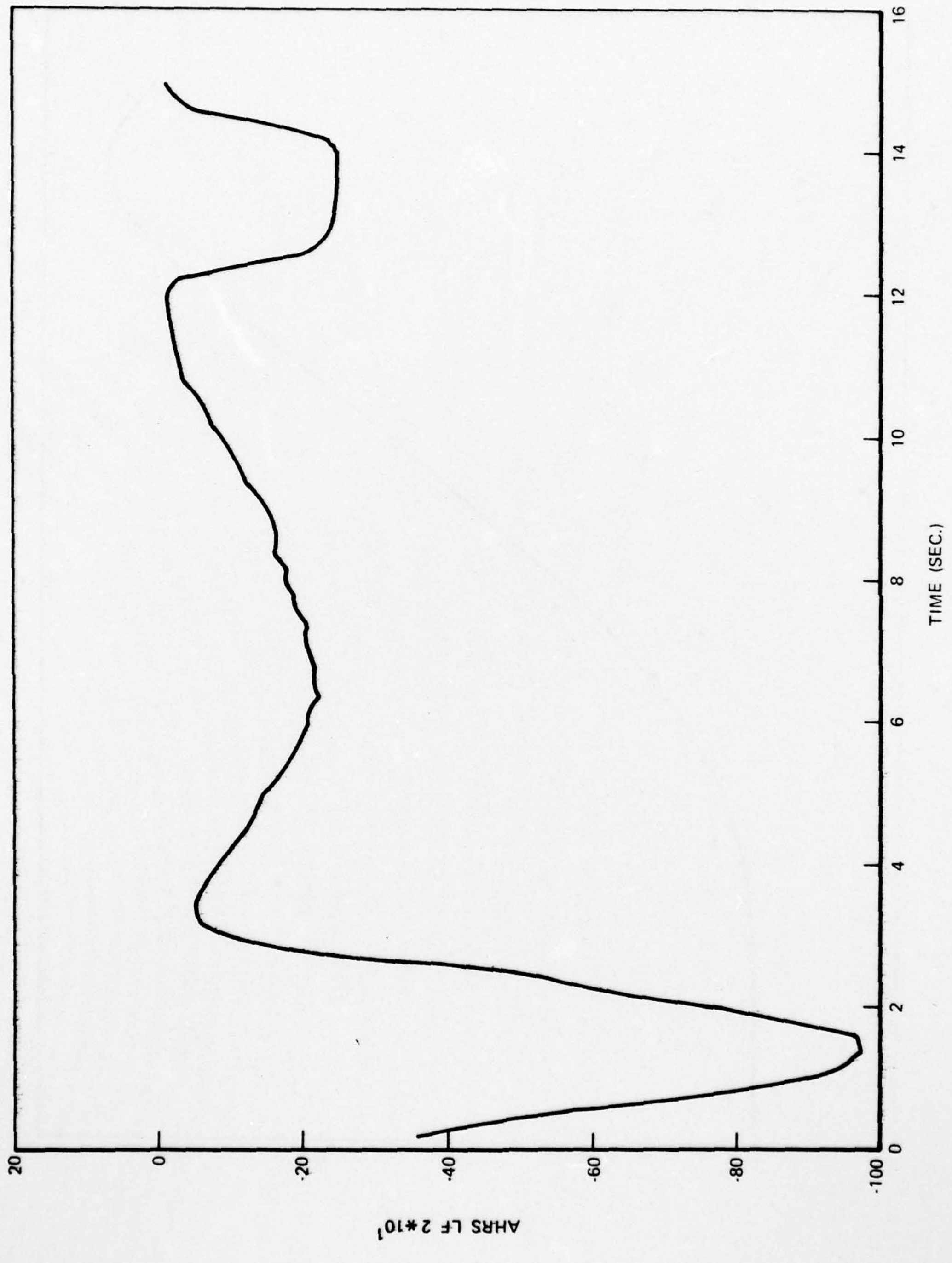


Figure 23e. Yaw Rate Gyro Failure

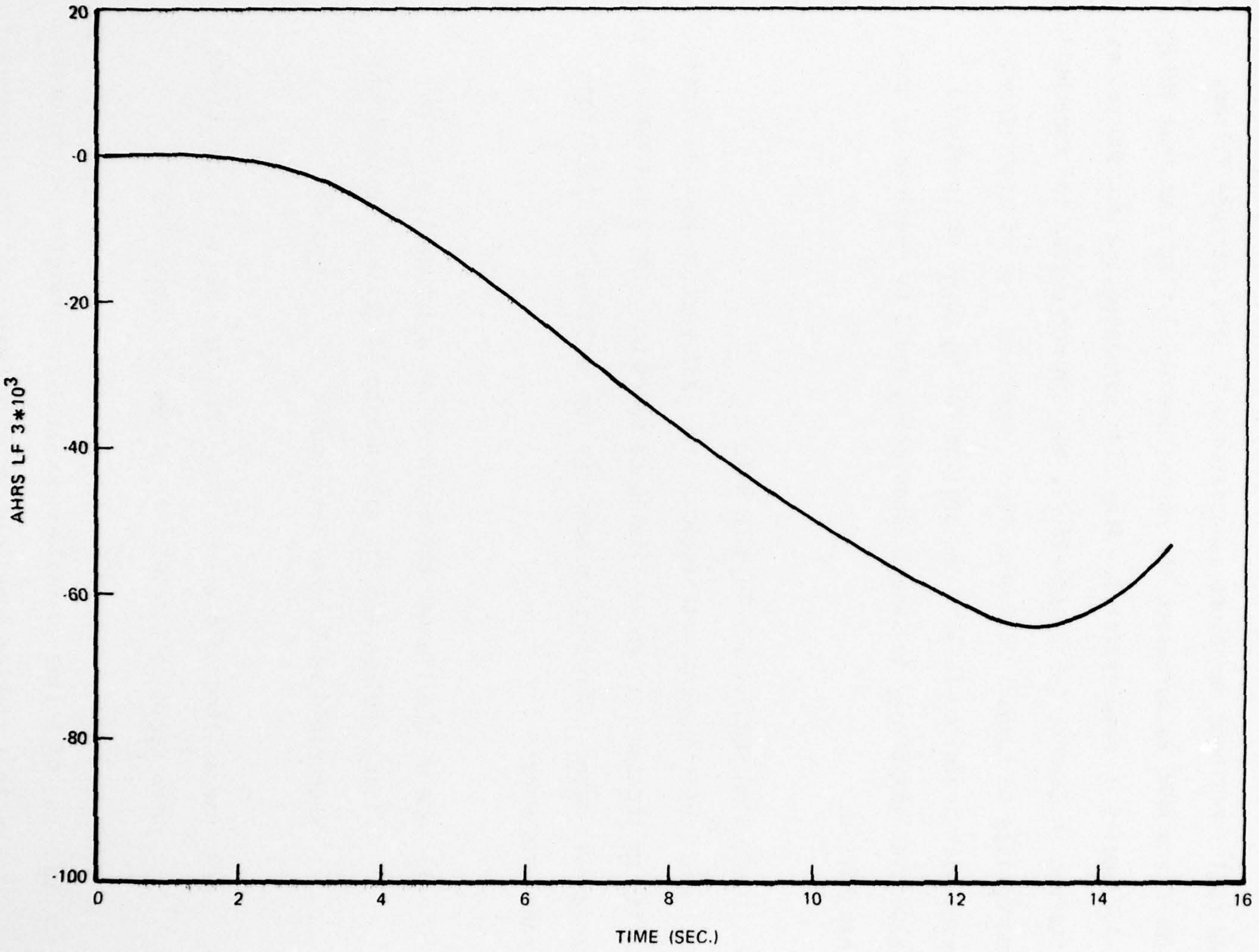


Figure 23f. Yaw Rate Gyro Failure

Figure 24 presents a typical result of failing the roll rate gyro in level flight, an experiment not reported in Table II. As expected, the roll likelihood functions associated with both attitude filters (INS 2 and AHRS 2) surpassed the no-failure level flight maximum values (-4.2 and -9.0, respectively). They also surpassed the maximum values attained in descent (-6.9 and -15.7), but did not exceed the thresholds appropriate to turns (-20.4 and -36). Therefore, use of tight thresholds with inhibited failure declaration during turns or threshold values set adaptively to sensed maneuvering would be required to detect this failure.

4. LESSENING SENSITIVITY TO ROLL RATE

One consistent characteristic in the last section was the sensitivity of the two roll angle likelihood functions (INS 2 and AHRS 2) to rapid roll rates. The three procedures for handling the resulting transients were:

(1) use of the tighter threshold values appropriate to other flight regimes and the disenabling of failure declarations when rapid roll rates were sensed (or commanded)

(2) use of threshold values that would be adaptively set higher when rapid roll rates were sensed (or commanded)

(3) use of a time-to-failure-declaration parameter in conjunction with a tighter threshold value to "mask out" the transient effects.

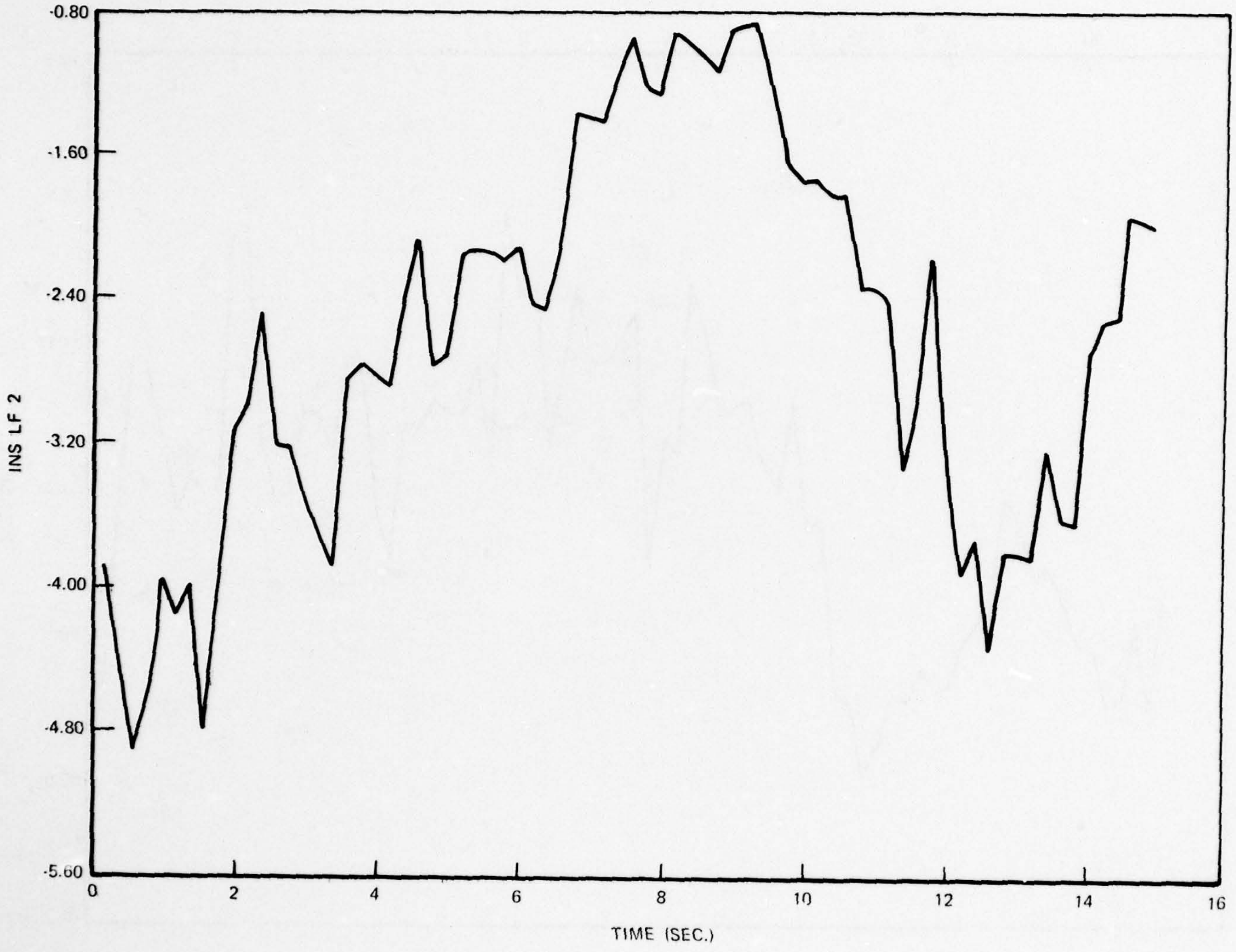


Figure 24a. Roll Rate Gyro Failure

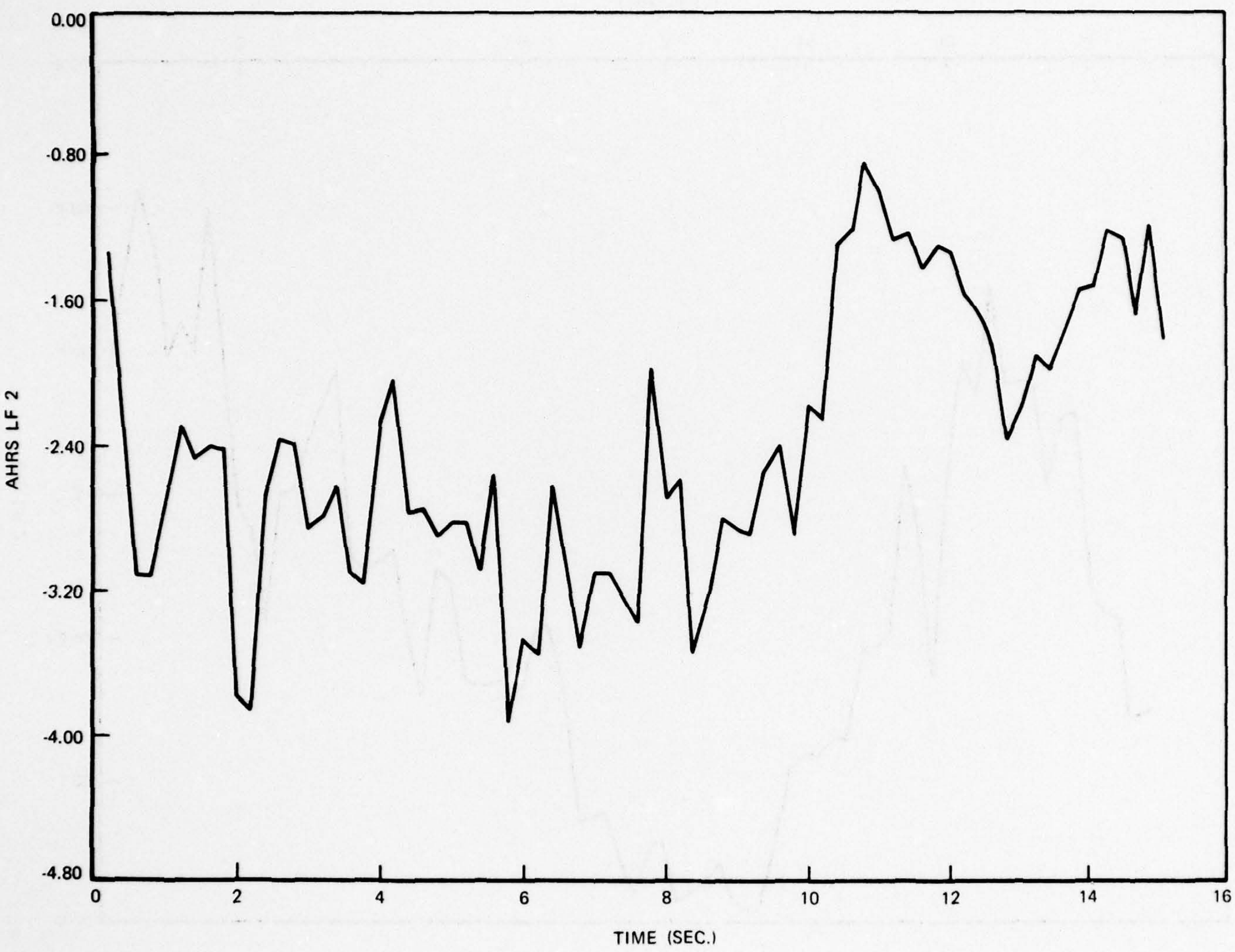


Figure 24b. Roll Rate Gyro Failure

Because of the magnitude of these transient effects, an investigation was made to determine if increasing the two filters' driving noise covariance matrix ($\underline{Q}$) components could significantly reduce the transient amplitudes, while leaving failure characteristics unchanged (or even enhanced).

The first tests simply multiplied the diagonal $\underline{Q}$ matrices by constant factors. This led to somewhat decreased transient magnitudes, but also to a decrease of likelihood function growth due to real failures.

Subsequently, it was reasoned that the real system model uncertainty was in the roll channel, so only the element in the first row and first column of the $\underline{Q}$ matrices (corresponding to driving noise on roll rate) was increased, leaving the other terms unchanged. Tests were conducted with these elements set to 2, 5, and 10 times the value that tuned the filters to the straight-and-level flight regime. Monte Carlo runs of turns were then made for (1) no failures simulated, (2) the yaw rate gyro failed, and (3) loss of vertical gyro cutoff. Since the two failure cases are indicative of the decrease in transient amplitude achieved in the no-failure case, only these plots are included here.

Figure 25 displays typical plots of INS 2 and AHRS 2 for the rate gyro failure case with Q_{11} set at twice its normal value. The transient is decreased somewhat from that depicted in Figure 23, and the intermediate growth is somewhat better as well. The trends of the other four likelihood functions were left unchanged, and failure detection was readily possible.

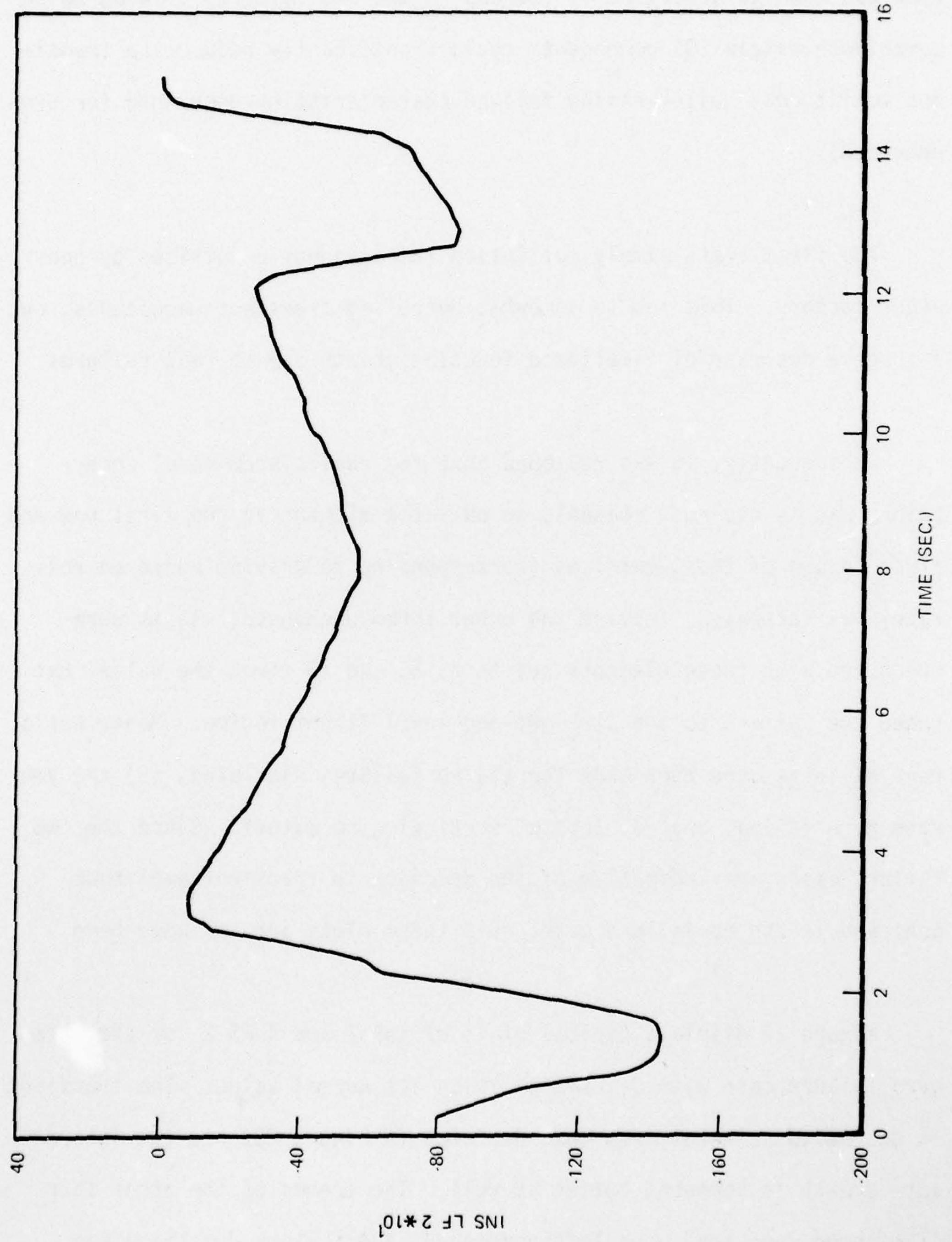


Figure 25a. Rate Gyro Failure; Q_{11} = Twice Normal Value

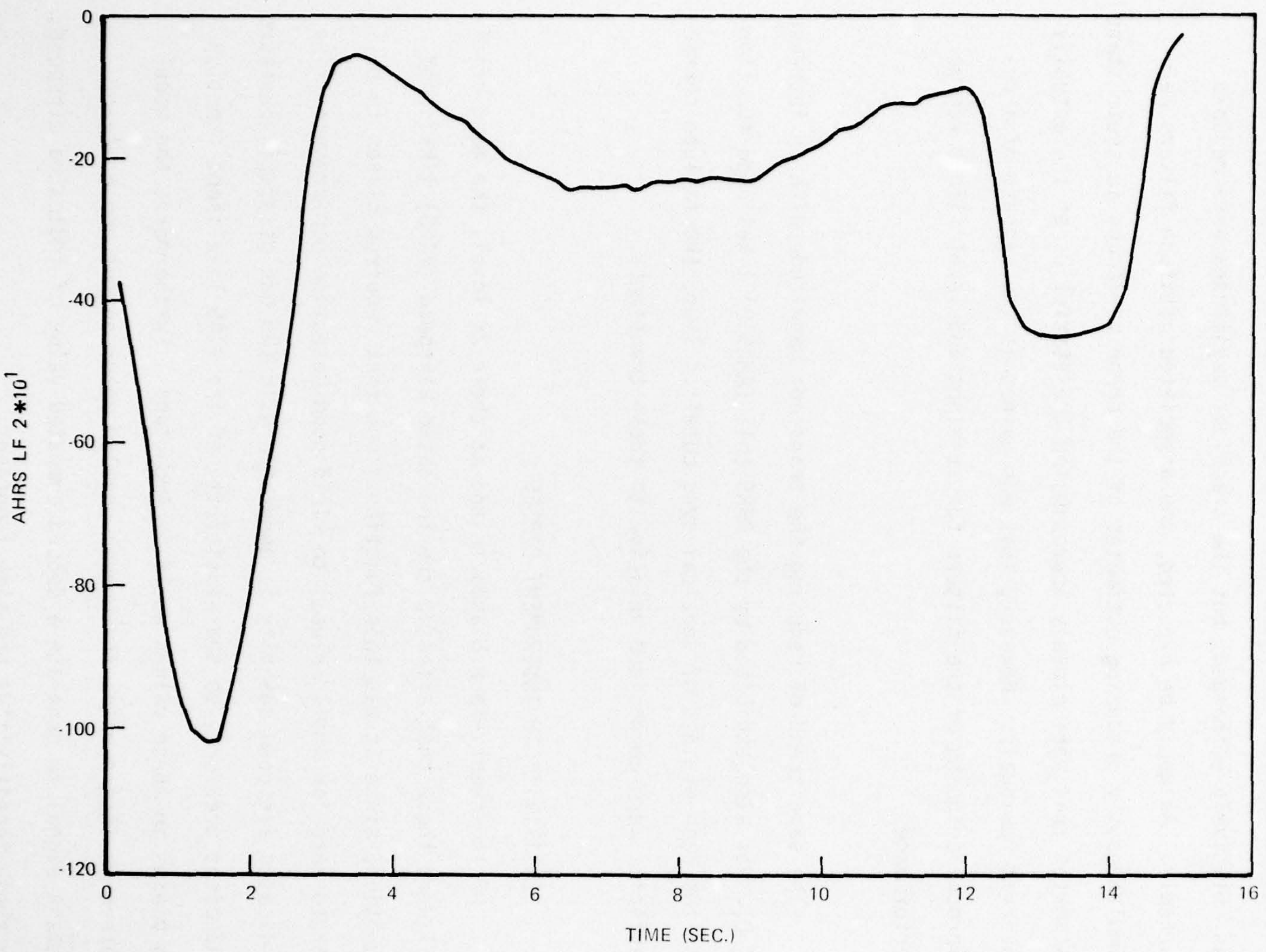


Figure 25b. Rate Gyro Failure; Q_{11} = Twice Normal Value

By increasing Q_{11} to ten times its normal value, results were such as those displayed in Figure 26. The growth due to the actual failure was relatively unchanged, but the transient magnitudes were reduced markedly. As would be expected, the associated attitude filters were simultaneously producing estimates of the error variances in their state estimates that more closely approximated the statistics of the actually observed residuals. However, this was gained at the expense of a reciprocal detuning of the filters for straight-and-level flight regime performance.

The same trend of reducing the transient magnitude with an increase of Q_{11} was also exhibited by the AHRS roll (AHRS 2) likelihood function for the case of loss of vertical gyro cutoff. Thus, the failure characteristics were emphasized relative to these transients.

5. SENSITIVITY TO INSTRUMENT BIASES

Of the twenty-four biases tested at their 2σ level, the most critical were those that affected the indicated airspeed (VIAS) likelihood function, since it was this function alone that required biases to be set to zero (or small values) to yield good detection performance. The indicated airspeed quantity is generated with the use of static pressure and pitot pressure, so the sensitivity of the VIAS likelihood function to biases in these values would be important. Furthermore, the model reference incorporates the normal accelerometer output and angle-of-attack signal to generate a second computed value of indicated airspeed, so these sensitivities are also important.

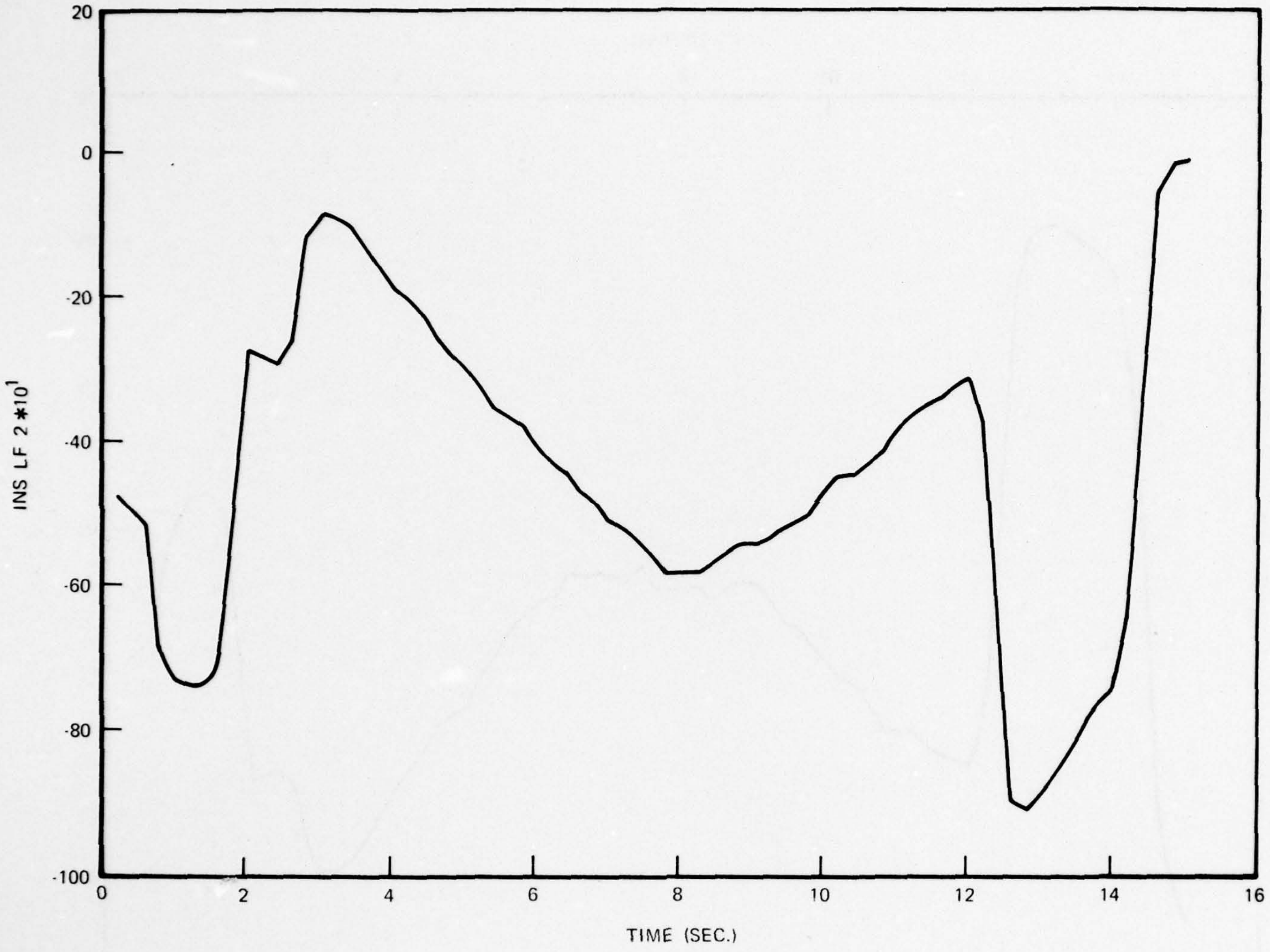


Figure 26a. Rate Gyro Failure; Q_{11} = 10 x Normal Value

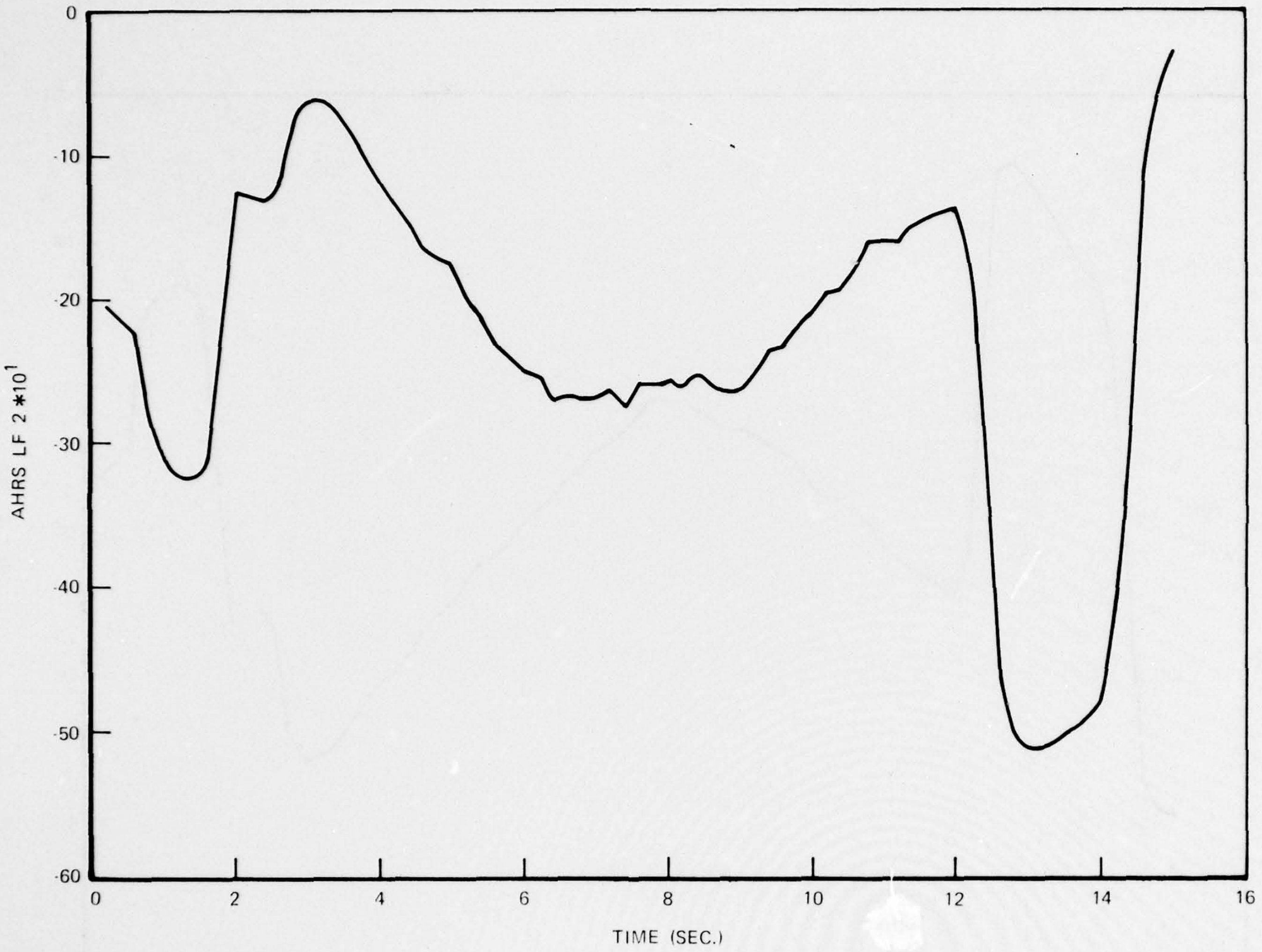


Figure 26b. Rate Gyro Failure; Q_{11} = 10 x Normal Value

Figure 27 presents the typical VIAS likelihood function in the level, turn, and descent trajectories during simulations in which the static pressure bias was set at its 2σ value. Similarly, Figure 28 portrays a typical example when the pitot pressure bias is similarly set. The extreme sensitivity of likelihood function performance to these two biases, as seen by comparing these plots to those of Figure 8, emphasizes the importance of removing such biases to the greatest possible extent during preflight, if this likelihood function of the detection logic is to perform properly. The angle-of-attack bias affects this likelihood function to a minor degree, changing its magnitude by about 5 to 10 at most when the bias is at its 2σ value. It should also be noted that the pitot pressure bias also affected the vertical filter, with the VERT 2 likelihood function doubling its threshold in level flight.

Other bias sensitivities are less critical to performance, and they were also found to be less severe than the static and pitot pressure bias sensitivities. Biases directly on the INS outputs caused some performance change. The 2σ bias on INS pitch caused the corresponding likelihood function to achieve maximum magnitudes of -7.5 in level flight, -4.3 in a turn, and -8.7 in descent (compared to the values -4.2, -2.04 and -6.9, respectively in Table V); the roll bias caused the roll likelihood function to reach -9.1 in level flight (compared to -5.7 from Table V). Other bias effects in the INS, as due to gyro or accelerometer biases, were negligible.

The AHRS biases directly on the Euler angle outputs similarly had some effect on likelihood performance. The pitch likelihood function

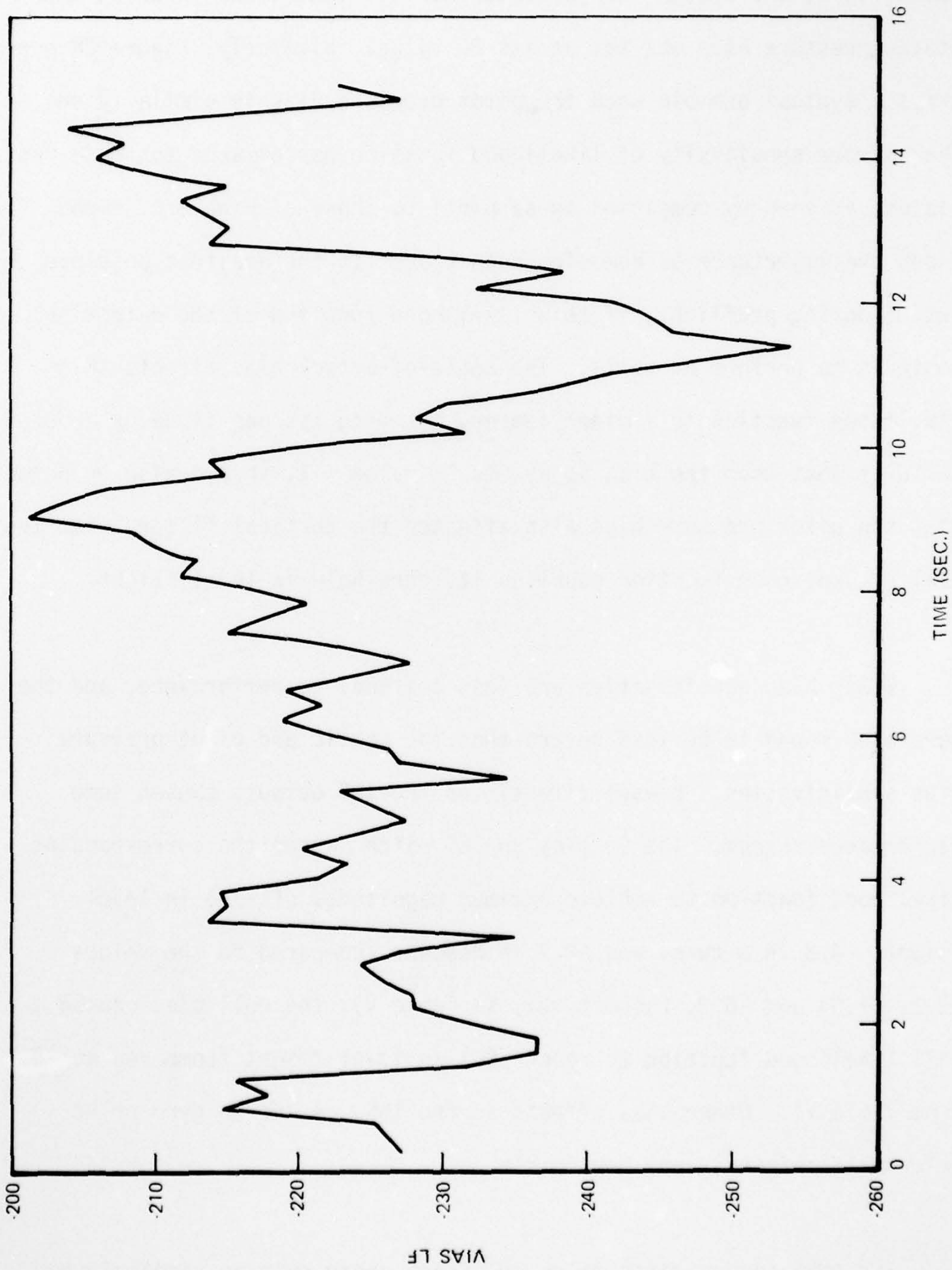


Figure 27a. Static Pressure Bias (Level Flight)

AD-A076 906 AIR FORCE FLIGHT DYNAMICS LAB WRIGHT-PATTERSON AFB OH F/G 1/3

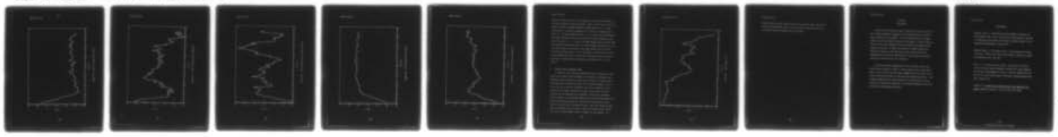
FAILURE DETECTION THROUGH FUNCTIONAL REDUNDANCY.(U)

SEP 76 P S MAYBECK

UNCLASSIFIED AFFDL-TR-76-93 NL

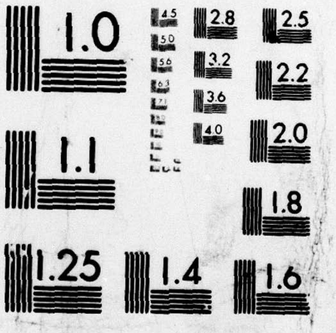
3 OF 3

AD
A076906



END

DATE
FILMED
12-79
DDC


1.0
4.5
5.0
5.6
6.3
7.1
2.8
3.2
3.6
4.0
2.5
2.2
2.0
1.1
1.8
1.25
1.4
1.6

X

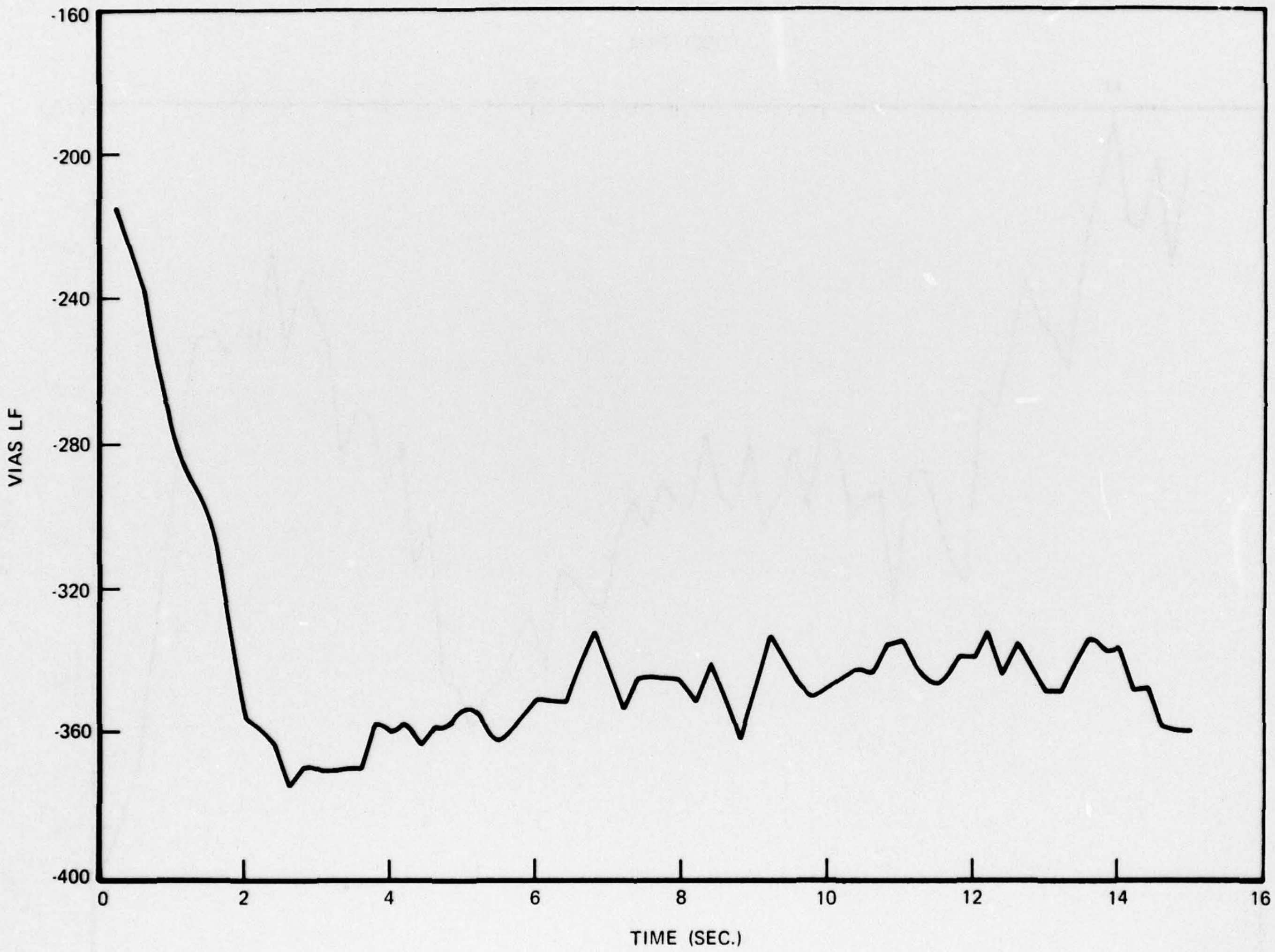


Figure 27b. Static Pressure Bias (Turn)

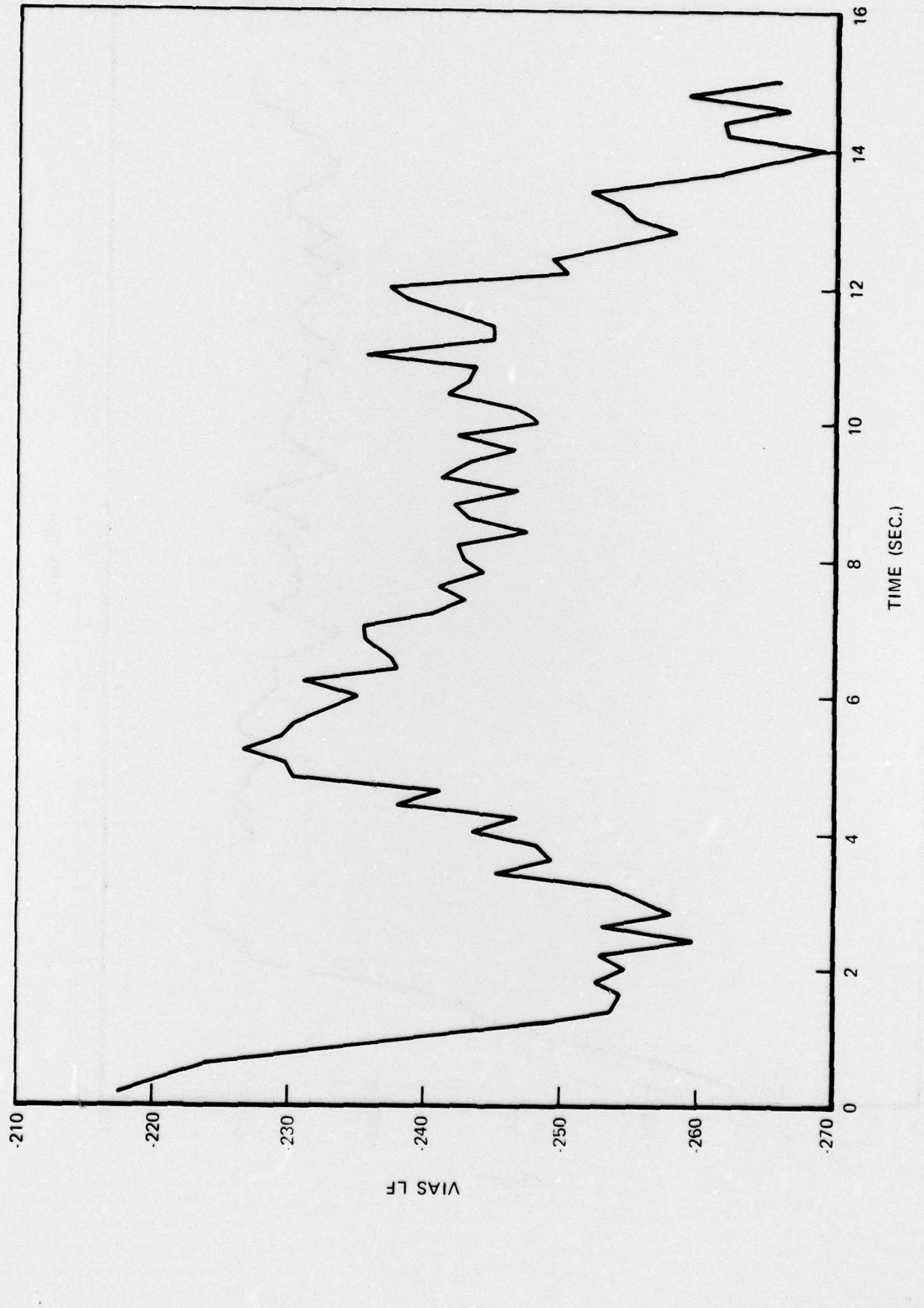


Figure 27c. Static Pressure Bias (Descent)

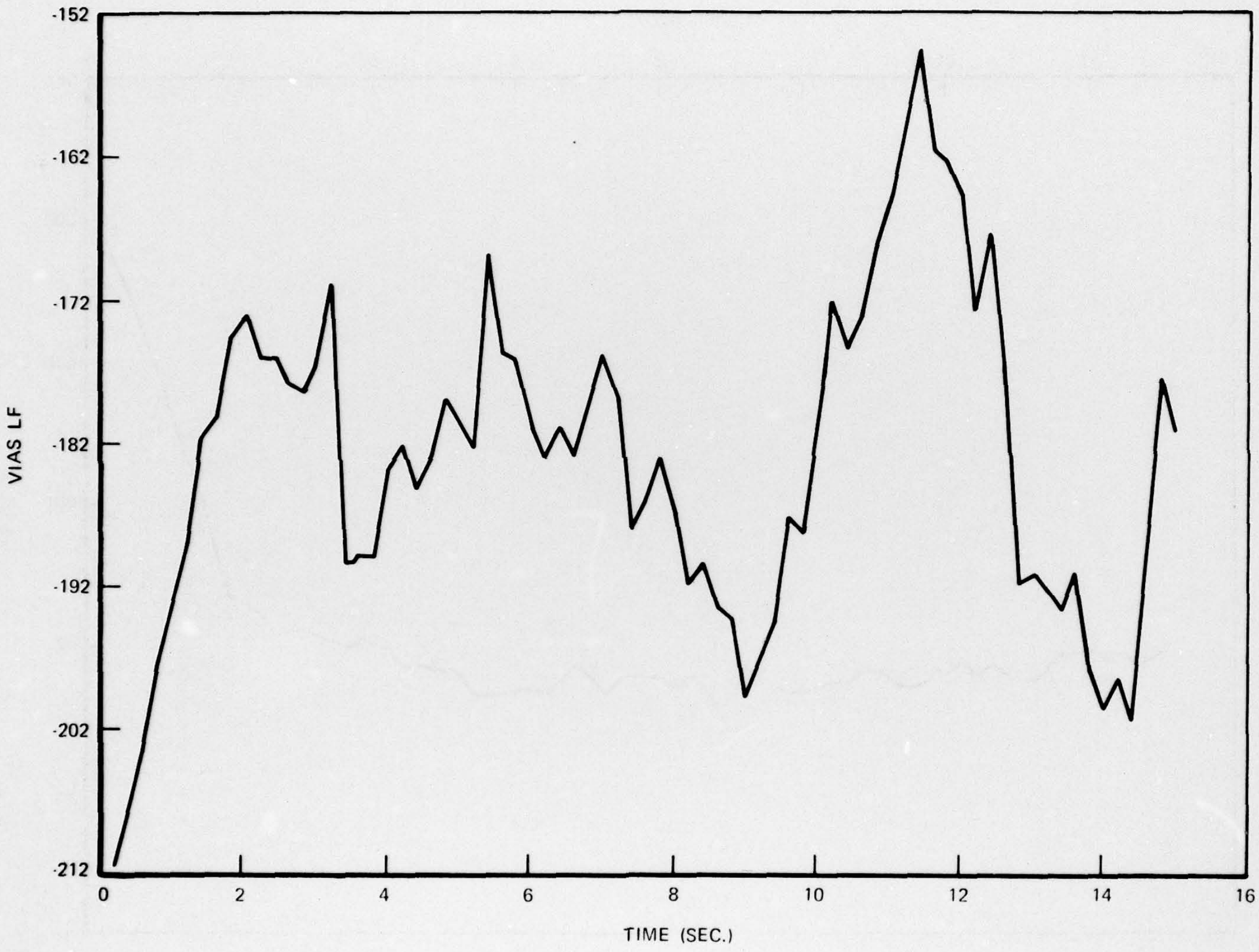


Figure 28a. Pitot Pressure Bias (Level Flight)

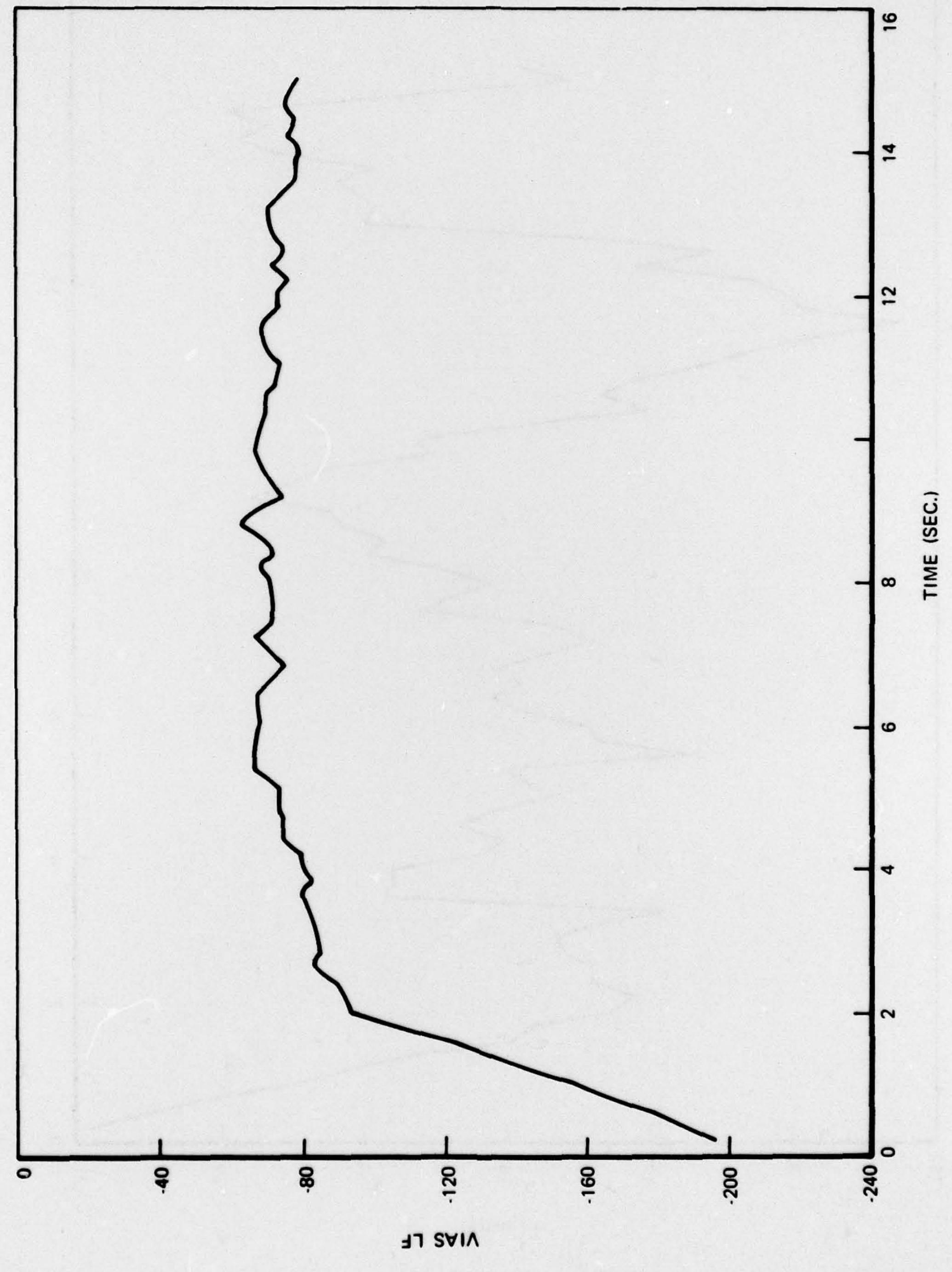


Figure 28b. Pitot Pressure Bias (Turn)

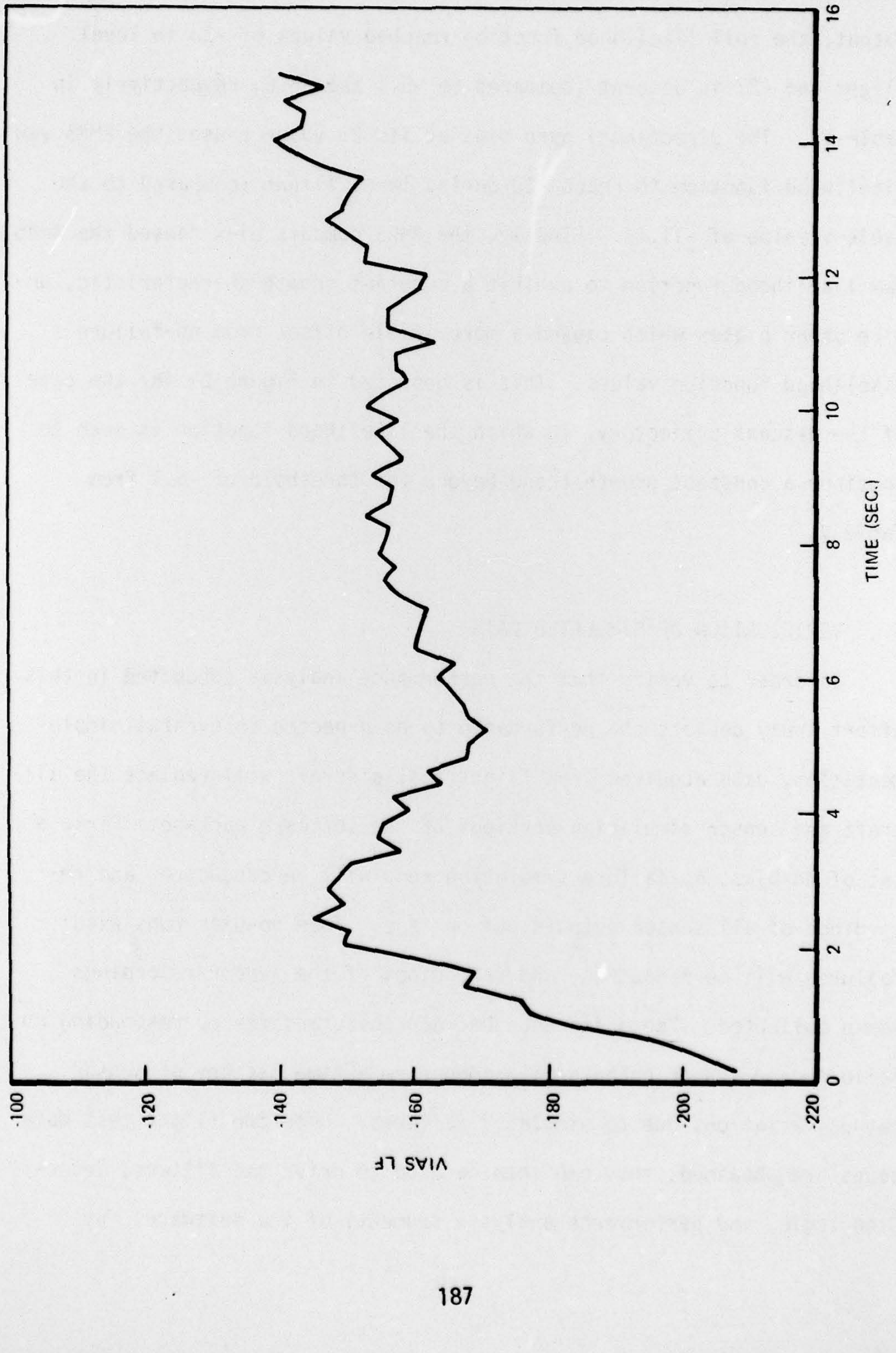


Figure 28c. Pitot Pressure Bias (Descent)

reached a value of -52 in a turn (compared to -36 in Table V) when a 2σ bias was simulated on the pitch output. With a similar bias on the roll output, the roll likelihood function reached values of -23 in level flight and -27 in descent (compared to -5.1 and -8.6, respectively in Table V). The directional gyro bias at its 2σ value causes the AHRS yaw likelihood function to reach -20 during level flight (compared to the Table V value of -11.1). Finally, the AHRS compass bias caused the AHRS yaw likelihood function to exhibit a constant growth characteristic, unlike other biases which caused a more stable offset from no-failure likelihood function values. This is depicted in Figure 29 for the case of the descent trajectory, in which the likelihood function is seen to continue a constant growth trend beyond the threshold of -6.1 from Table V.

6. VERIFICATION OF SIMULATED DATA

In order to verify that the performance analysis conducted in this effort truly depicts the performance to be expected in eventual implementation, data acquired from flight test aircraft will replace the aircraft and sensor simulation portions of the software package. First a set of no-bias, no-failure simulation runs will be conducted, and recordings of all sensor outputs put on tape. Then no-bias runs with failures will be conducted, and recordings of the sensor recordings again collected. The difference between these and the corresponding no-failure case sensor outputs will then form a time history of sensor output variations due to simulated failures. Once the flight test data tapes are obtained, they can then be used to drive the filters, detection logic, and performance analysis segments of the software. By

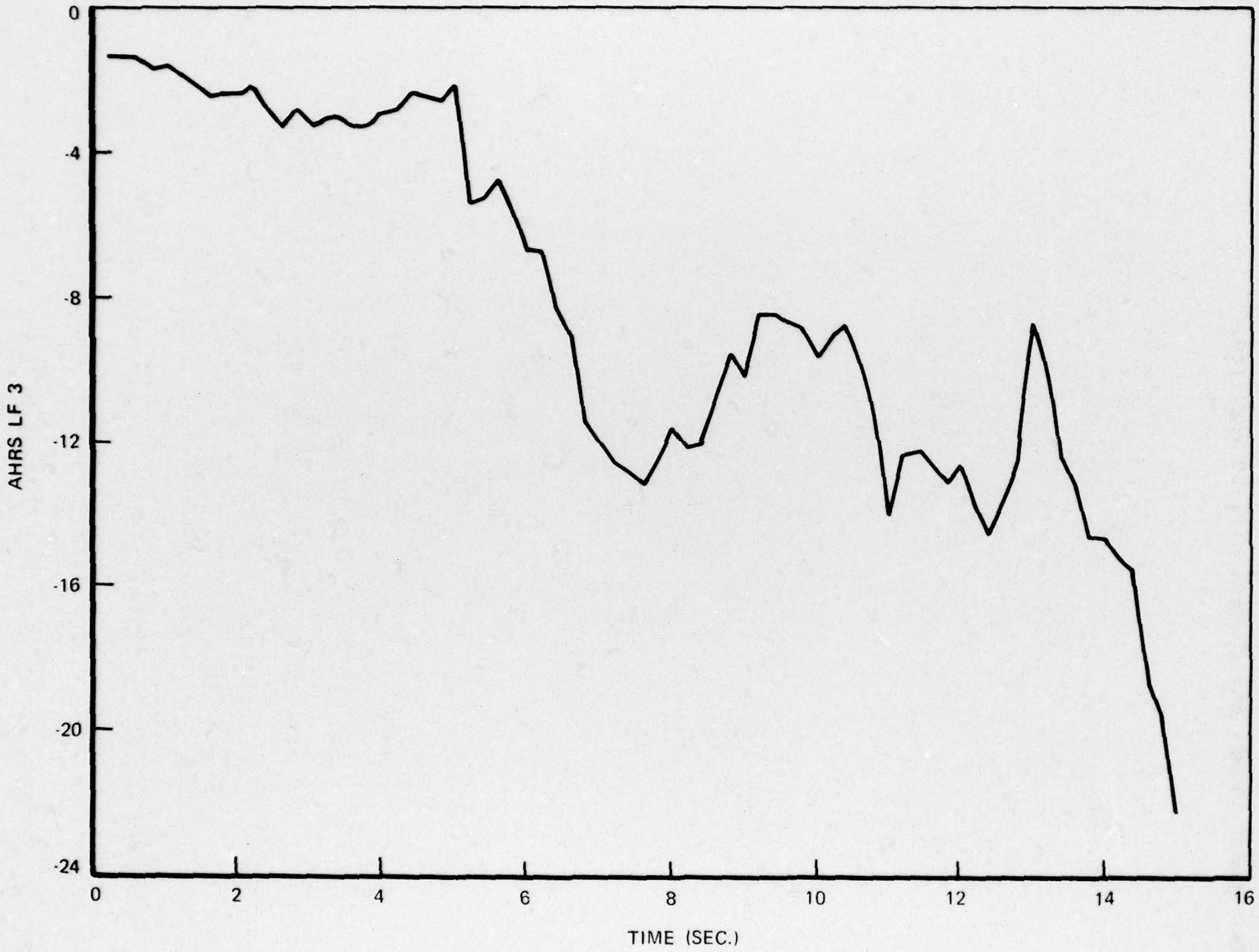


Figure 29. AHRS Compass Bias

superimposing the sensor output variations on the real data, the ability of the failure detection logic to discern failure characteristics in a real sensor signal environment will be verified.

SECTION V

CONCLUSION

An efficient and effective means of detecting failures of data sensors through functional redundancy has been developed and its performance capabilities investigated. As the preceding section has demonstrated, the failure detection power of the concept is rather extensive. Since it allows such detection by combining data from systems already onboard an aircraft, it reduces the amount of hardware duplication required to achieve a specified level of data system reliability. Consequently, the practical implications in cost, weight, and volume savings for future aircraft are substantial.

To aid the eventual implementation of this concept into an integrated failure detection system, a flexible design tool has been developed. With this tool, the functional redundancy detection logic can be readily tuned and optimized for any particular onboard application. The software package that has been developed can significantly assist the conversion of performance potential of functional redundancy failure detection into performance realization.

BIBLIOGRAPHY

Maybeck, Peter S., "Combined State and Parameter Estimation for Online Applications," Ph.D. dissertation at Massachusetts Institute of Technology (also Charles Stark Draper Laboratory Report T-557), Cambridge, Massachusetts, February 1972.

Maybeck, Peter S., "The Kalman Filter - An Introduction for Potential Users," TM-72-3, Air Force Flight Dynamics Laboratory, Wright-Patterson AFB, Ohio, June 1972.

Meier, L., D.W. Ross, and M.B. Glaser, "Evaluation of the Feasibility of Using Internal Redundancy to Detect and Isolate Onboard Control Data Instrumentation Failures," Technical Report AFFDL-TR-70-172, Air Force Flight Dynamics Laboratory, Wright-Patterson AFB, Ohio, January 1971.

Papoulis, A., Probability, Random Variables, and Stochastic Processes, McGraw-Hill Book Co., New York, New York, 1965.

*U.S.Government Printing Office: 1979 — 657-002/193